Naval Command,
Control and Ocean
Surveillance Center    RDT&E Division

San Diego, CA
92152-5001

# AD-A266 101

‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖

Technical Document 2515
May 1993

# Data and Knowledge Engineering (DKE)

Case Study Report

BBN Systems and Technologies

93-14247

‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖

93-14248

93  6  24  010

NRaD

Technical Document 2515
May 1993

# Data and Knowledge Engineering (DKE)

## Case Study Report

BBN Systems and Technologies

## NAVAL COMMAND, CONTROL AND OCEAN SURVEILLANCE CENTER RDT&E DIVISION
## San Diego, California 92152–5001

**J. D. FONTANA, CAPT, USN**
**Commanding Officer**

**R. T. SHEARER**
**Executive Director**

## ADMINISTRATIVE INFORMATION

PK

# Table of Contents

# List of Figures

UNCLASSIFIED

## Purpose

The purpose of this case study is to identify potential for the application of Active Database (ADB) concepts to Navy Command and Control ($C^2$).

## Overview

Military $C^2$ databases are in the process of transitioning to relational database management systems (RDBMS). This will greatly enhance the user's ability to query the databases and extract useful information rapidly, however it will not address the existing requirements for data integrity and consistency or situation monitoring.

Traditionally, DBMS are passive, they execute queries or transactions only when explicitly requested to do so by a user or application program.[1] This is particularly true of both existing Navy $C^2$ databases and the RDBMSs that are replacing them. As a result, large, complex, and cumbersome applications have been developed to process incoming data, compose and execute database update transactions, and query the database. The transition to RDBMS may make it easier to develop applications which update, query, or conduct consistency checks on the database, but they will still be external applications. Comprehensive data consistency checks will have to be conducted at some specified interval and require a variety of complex database queries which will have to be analyzed and compared in order to identify possible inconsistencies. Between data consistency checks, the database will contain a variety of internally inconsistent data which may complicate or invalidate any decision on which the data is based.

An active (or reactive) database could be capable of performing many of the functions currently performed by external applications in a manner more congruent with maintaining internal database consistency and alerting users to situations which may require intervention. The update process of an active database could include complete data consistency checks, inconsistent data could be corrected automatically (e.g. unit identification errors that could be resolved internally) or referred to an operator along with a body of supporting data which would allow the operator to better resolve the inconsistency.

---

[1] S. Chakravaíthy, "Active Database Management Systems: Requirements, State-of-the-art, and an Evaluation". University of Florida, Gainsville, FL, 1991.

Functionally, an active database management system monitors conditions triggered by *events* representing database events (e.g., updates) or non-database events (e.g., events detected by an external application) and if the *condition* evaluates to true then an action is *executed.*[2]

## Navy Command and Control

Navy $C^2$ is a data critical function of Naval Warfare. In the past decade Navy $C^2$ information management has partially transitioned from a variety of hierarchichal, proprietary, and flat file databases to relational database management systems. At the same time, the volume of data that is received, stored, and requires analysis has expanded dramatically. During the next decade the transition to RDBMS will be completed.

A simplified model of the inputs and outputs of a Navy $C^2$ system are shown in Figure 1. Inputs consist of a variety of formatted reports received from external sources, data updates from other $C^2$ computers systems, and operator inputs. Outputs include updates to other $C^2$ computers and various reports.

External events comprises any and all occurrences which are of interest to a Navy decision maker. Some of these activities result in formatted reports which are processed as database updates, e.g. Status of Operational Readiness and Training (SORTS) reports. Comprehensive updates to the core (static) data of the database (e.g. unit characteristics and performance data) are performed at specified intervals as updates to the originating databases (e.g. Naval Warfare Tactical Database NWTDB) are received.

A significant number of external events occur for which no appropriate database transaction exists. These can be categorized as "situation" reports which appraise the chain of command of changes in the situation which may require action, but contain no specific data which can be recorded in a database. Situation reports often trigger the development of contingency plans. The development of a contingency plan seldom results in changes to the C2 database. The successful implementation or execution of a contingency plan relies heavily on the data contained in the database and does affect wholesale changes to the database. The development of plans in response to a perceived change in the current situation can be referred to as "situational planning".

---

[2] IBID.
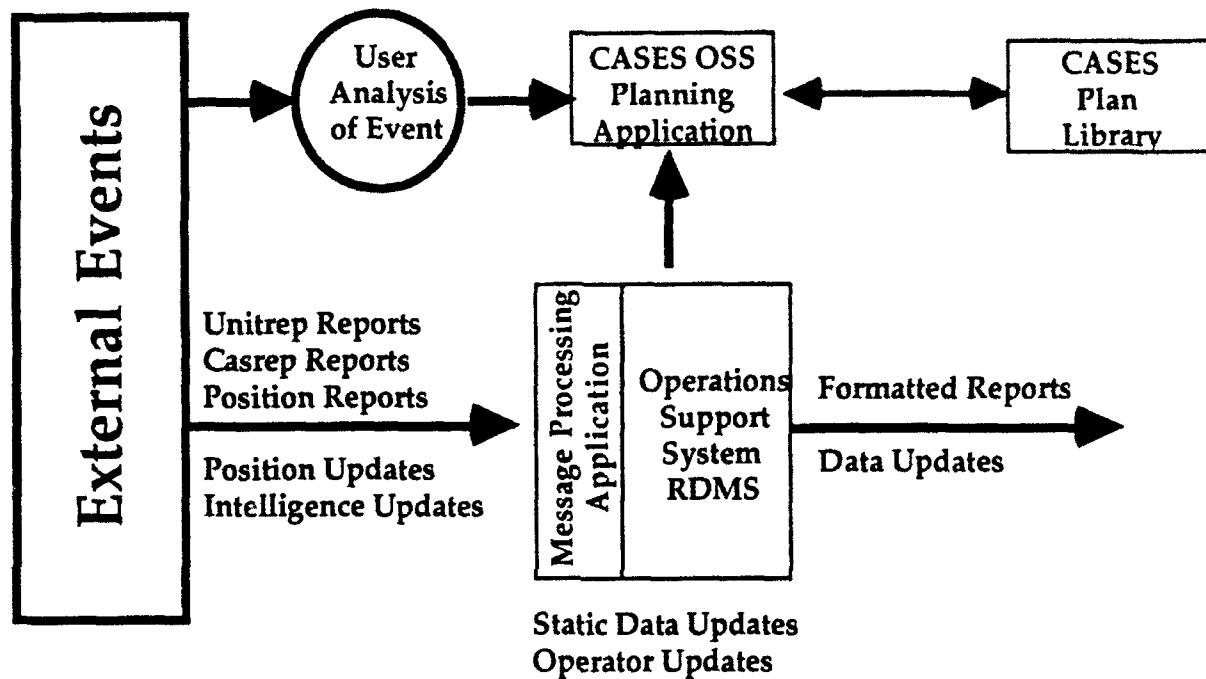
Figure 1 Simplified View of OSS Data Flows

In the following sections various potential applications of ADB concepts within the C2 arena are examined. The potential for application of ADB concepts to C2 databases is certainly not limited to the examples below.

## DB Consistency/Integrity

Data consistency and integrity is critical to the reliability and credibility of the database and consequentially it's use in decision making. The decision making process is significantly affected by the perceived credibility of the data available. There are four classic "error" conditions from the world of probability which can be used to describe the impact of the database on decision making process.

| A | Event Detected<br><br>Event Present | C | Event Detected<br><br>Event NOT Present |
|---|---|---|---|
| B | Event NOT Detected<br><br>Event Present | D | Event NOT Detected<br><br>Event NOT Present |

**Figure 2 Classic "Error" States from Probability Theory**

A & D define the two sides of a simple probability, if A then the event occurs or is TRUE; if D then the event does not occur or is FALSE. B & C define the probabilities of "false alarm". These additional probabilities are necessary to define the full range of conditions in the real world, i.e. something happens, but is not observed or an apparent observation turns out to be invalid.

The decision making process always starts with an underlying assumption concerning of the credibility of the data available. This can be represented by an analogy of the probability error states shown in the following figure.

| 1 | Data perceived valid<br><br>Data VALID | 3 | Data perceived valid<br><br>Data NOT VALID |
|---|---|---|---|
| 2 | Data perceived NOT valid<br><br>Data VALID | 4 | Data perceived NOT valid<br><br>Data NOT VALID |

**Figure 3 Analogy of "Error" States in Decision Making Databases**

In reality, in a complex decision making environment such as Navy $C^2$, which relies on large aggregations of data from a wide variety of sources, all four conditions can be present in any situation. It is vitally important to the decision making process that conditions 2 & 3 be minimized. ADB concepts seems ideally suited to many of the data consistency issues.

## Updates to Readiness Databases

The World Wide Military Command and Control System (WWMCCS) maintains a complete database on the current status of each military unit. The individual unit is responsible for updating the database as changes occur in it's readiness state. In order to minimize the volume of communications, only changes are communicated in each SORTS report.

```
Fm : USS Neversail
To: WWMCCS

//UNCLASSIFIED//
//Neversail/CG/N52505//
//SORTS/0204//
//9202231600Z//
//CROVL/5/920317/1/EKG/EKI/PCH//
//CEQPT/5/EKG/EKI//
//MOB/5/920317/2/EKG/PCH//
//ACTIV/INPORT//
//ENDAT//
```

**Figure 4 Sorts Report Example**

The WWMCCS software is capable of ensuring that all previous reports have been received and changes applied before applying new changes. It is capable of detecting transmission or format errors which might invalidate the message. The current software is also capable of detecting obvious "range or domain" errors. In all cases, the only response the WWMCCS software is capable of is "rejecting" the message into an error queue where the message is manually processed or returned to sender. The percentage of WWMCCS update messages which fail automatic processing due to these basic error checking routines is often extremely high which results in a large number of messages being manually processed and a large number of messages rejected back to the reporting unit. In many situations, these errors could be automatically corrected. The following is an example:

> A unit reports it's readiness through the assignment of a number, a 1 indicates the unit is completely ready to accomplish it's assigned missions and a 4 indicates the unit is not capable of accomplishing it's assigned mission; 2 and 3 are intermediate values. A readiness rating of 5 is reserved for units which are undergoing "scheduled" major maintenance. These units are known not to be mission capable, but are not "counted" since they are undergoing maintenance. There are only specific categories of activity which a unit can be assigned to and report a readiness of 5. If WWMCCS receives a report in which a unit reports a readiness of 5, but does not correctly report an appropriate activity, WWMCCS rejects the message. The example SORTS report would be rejected.

In the example cited above, there is data within the same database which could be used to "check" and possibly correct the incoming report by applying ADB concepts.

| Event | Condition | Action |
|---|---|---|
| SORTS Report Received | If Reported Overall Readiness = 5<br><br>and<br><br>If reported Activity (e.g. INPORT) not consistent with Readiness of 5 | a) Compare scheduled activity with reported activity.<br><br>b) If scheduled for maintenance:<br>  1) update Activity<br>  2) Alert operator<br>  3) Advise unit<br><br>c) If not scheduled for maintenance:<br>  1) reject report.<br>  2) Alert operator<br>  3) Advise unit |

Figure 5 ECA SORTS Reported Readiness vs. Activity

The opposite situation is also common, a unit is undergoing scheduled maintenance, but continues to report it's readiness in the range of 1 to 4. WWMCCS does not detect this condition. In this case the ADB event could be temporal based vice event based.

| Event | Condition | Action |
|---|---|---|
| Daily at 0400 GMT | If reported readiness and activity are inconsistent with scheduled activity. | a) Check "currency" of readiness data.<br><br>b) Check "currency" of schedule data.<br><br>c) Advise operator of potential inconsistency. |

Figure 6 ECA SORTS Temporal Based Event

In either case the database is both internally inconsistent and "out of sync" with the current situation in the real world and could impact the decision making process. It is probably not a good idea rely on any system to automatically correct critical data without human supervision, however reviewing corrective actions should be a lot more efficient than manually researching and applying changes and a lot more timely than the process of rejecting a message and waiting for a corrected message to be composed, approved, transmitted, received and processed.

*Consistency across warfare and resource areas*

Readiness reporting consists of a matrix of warfare areas and resource areas as shown in the following example. The assignment of specific ratings in each resource area is governed by sets of rules. In most cases the rules are very specific and require the completion of complex worksheets to determine the current readiness status. There is some leeway for a commander's subjective opinion as can be seen in the two Warfare ratings and two Resource Area ratings where there are two possible readiness ratings. Anti-Air Warfare (AAW) is a good example of where both objective rules and subjective judgment apply. The objective rule is that the overall warfare rating cannot be higher than the higher of the two lowest ratings, in this case it cannot be 1, but can be either 2 or 3. The commander is allowed to make a subjective judgment whether the rating is reported as a 2 or a 3. There can be many factors which affect the final determination. A ship which had only a AAW self-defense capability might report a readiness of 2, whereas a ship which was responsible for AAW defense of the battlegroup might consider this to be much more critical and report a readiness of 3.

| Resource Area<br><br>Warfare Area | Personnel | Training | Supplies | Equipmnt | Warfare<br>Rating |
|---|---|---|---|---|---|
| Anti-Air Warfare | 1 | 2 | 2 | 3 | 2 or 3 |
| Anti-Sub Warfare | 1 | 1 | 2 | 2 | 2 |
| Mobility | 1 | 1 | 1 | 2 | 1 or 2 |
| Resource Rating | 1 | 1 or 2 | 2 | 2 or 3 | Overall Rating<br><br>2 or 3 |

**Figure 7 SORTS readiness matrix**

Whenever a unit reports a degradation in Warfare Area readiness, it should also report a reason (coded to the applicable Resource Area), a change in the applicable Resource Area (if appropriate) and an anticipated date when the readiness will improve (or degrade further). Errors of both commission and omission often occur in the readiness reporting process which result in inconsistencies between a unit's Resource Areas, Warfare Mission Areas, and Reasons.

| Event | Condition | Action |
|---|---|---|
| SORTS Received | If reported readiness changes and reasons are not consistent with readiness database across both Resource Areas and Warfare Areas. | a) Advise operator of potential inconsistencies.<br><br>b) Update Resource Areas to be consistent with reported Warfare Areas |

**Figure 8 ECA SORTS Readiness Consistency**

Consistency between readiness and equipment casualty reports

Whenever a unit reports a degradation in readiness which is due to an equipment failure, it is also required to send a Casualty Report (CASREP). There is considerable overlap between SORTS and CASREPs, however the CASREP generally provides significantly more detail and also is supposed to trigger actions by the logistics chain to support the repair or replacement of the affected equipment. The data from both the SORTS and CASREPs are maintained in the same database. From the viewpoint of database consistency, the information contained in the readiness database must be consistent with the current outstanding CASREPs.

| Event | Condition | Action |
|---|---|---|
| SORTS Received | If degradation in Equipment Resource Area reported | a) Check for supporting CASREP. |
| CASREP Received | If degradation in specific equipment reported. | a) Check that readiness database is consistent. |

Figure 9 ECA SORTS/CASREP Consistency

It is also possible to imagine a much more active approach to maintaining the readiness database which would eliminate the requirement to file both a CASREP and a SORTS covering the same equipment failure.

| Event | Condition | Action |
|---|---|---|
| CASREP Received | If reported readiness changes and reasons are not included in readiness database. | a) Update readiness database. |

Figure 10 ECA CASREP Automatic Update of Readiness Database

Updates to CASREPs are submitted for a variety of reasons including changes in the estimate to correct the casualty, receipt of parts, correction of the casualty (CASCOR) and cancellation of the CASREP (CASCAN) for reasons other than repair of the casualty. A CASCAN might be filed if the broken equipment was removed from the ship's required

capability or replaced as the result of an upgrade during overhaul. Each these reports could require an update in the readiness database and result in the separate submission of a SORTS report. An active database could easily handle these "administrative" updates with a concomitant reduction in message traffic and database inconsistencies.

| Event | Condition | Action |
|-------|-----------|--------|
| CASREP Update Received | If estimated time of repair changes. | a) Update readiness database. |
| CASCOR Received | If reported readiness changes. | a) Update readiness database. |
| CASCAN Received | If reported readiness changes. | a) Update readiness database. |

Figure 11 ECA CASREP Updates

Consistency between equipment installed and equipment reported

There is a direct connection between the equipments installed or possessed by a unit and it's warfare capabilities. There is also an ongoing modernization program in the military which seeks to update current the capabilities or even add new capabilities to a unit. A good example is the current program to install vertical launch systems on Spruance class destroyers. Once installed this results in a new mission area for the unit. There are multiple parts of the database which are affected by the addition of this new capability. The readiness database must include the new mission area, the installed equipment database must be updated to reflect the new equipment, and the expendables database must be updated to show which types of missiles are authorized to be carried and how many of each. There is also a direct connection between the number of missiles carried and the maximum possible readiness in this area.

| Event | Condition | Action |
|---|---|---|
| SORTS Received | If a new mission area reported | a) Check installed equipment database. Report inconsistencies.<br><br>b) Check expendables equipment database for correct number of expendables for reported readiness. Report inconsistencies, if appropriate, update readiness database. |
| Update to Installed Equipment database received | If equipments related to specific mission area(s). | a) Check that readiness database includes mission area. Report inconsistencies. |

**Figure 12 ECA SORTS New Mission/Equipment**

| Event | Condition | Action |
|---|---|---|
| SORTS Received reporting change in number of expendables on board. | If number inconsistent with reported readiness. | a) Advise operator of inconsistencies.<br><br>b) Update Resource Areas and Warfare Areas to be consistent with reported numbers. |
| | If expendables reported are inconsistent with installed equipment. | a) Advise operator of inconsistencies. |

**Figure 13 ECA SORTS Change in Expendables**

## Updates to Positional Databases

Units are required to report their positions regularly via a variety of methods, both automatic and manual. In addition, units report the presence of other units. Different sensors and navigation systems provide

UNCLASSIFIED
11

significantly disparate positional information and a wide variety of errors, many unavoidable, can creep into the entire positional reporting process (e.g. own navigation errors, bearing or range errors, identification errors). Correlation algorithms are invoked to determine which position reports are valid. This is a very complex process and could easily be the entire focus of an ADB project. It is beyond the scope of this project to evaluate correlation algorithms.

Consistency between scheduled and reported locations

As previously stated, the WWMCCS database includes a schedule database which contains planned activities and locations for those activities along with start and end dates and other required information. Often the schedule database does not accurately reflect the current assignment of a unit or conversely, due to a variety of reasons, a unit's current location may not allow it to accomplish a scheduled assignment due to geographic constraints associated with the assignment.

| Event | Condition | Action |
|-------|-----------|--------|
| Position Report Received | If reported position and scheduled location are inconsistent. | a ) Report inconsistency. |

Figure 14 ECA Position Report vs. Scheduled Location

Updates to Schedule Databases

Feasibility of schedule changes

A schedule change must be feasible, e.g. it must be possible for the unit to accomplish the assignment. This ranges from geographic feasibility similar to those discussed in the previous paragraph to matching the units capabilities with the capabilities required by the assignment.

| Event | Condition | Action |
|-------|-----------|--------|
| Schedule Change Received | If change in current assignment or next assignment and it is not feasible for unit to travel the required distance. | a ) Report inconsistency. |

Figure 15 ECA Schedule Change Feasibility

Impact on future events

The ability of a unit to accomplish future assignments can be affected by schedule changes. Geographic feasibility has already been addressed, but start and end dates may overlap, additional fuel, expendables, or equipment may be required.

PERSTEMPO calculation

Changes in schedule can result in changes to a unit's PERSTEMPO or other measure of effectiveness (MOE).

| Event | Condition | Action |
|---|---|---|
| Schedule Change Received | If conditions of activity change, INPORT/AT SEA, length of assignment. | a) Recalculate MOEs. |

Figure 16 ECA Schedule Change Morale Impacts

Impact on Budget

Changes in schedule can result in changes to a budgeted cost (e.g. fuel budget, expendables budget).

| Event | Condition | Action |
|---|---|---|
| Schedule Change Received | If conditions of activity change, INPORT/AT SEA, length of assignment. | a) Recalculate fuel budget.<br><br>b) Report Changes. |
| | If expendables required changes | a) Adjust budget.<br><br>b) Report Changes. |

Figure 17 ECA Schedule Change Budget Impacts

## Situation Monitoring

Situation Monitoring supports decision makers by identifying and high-lighting changes in the current situation which may require action on the

part of decision makers to resolve. Accurate situation monitoring is dependent on the database consistency and integrity issues discussed in the previous section. Many of the following examples of situation monitoring were prototyped in the Force Requirements Expert System (FRESH) which was part of the Fleet Command Center Battle Management Program. FRESH demonstrated the utility of situation monitoring but was hampered by database consistency and integrity problems.

Updates to Readiness Databases

Changes in a unit's readiness can seriously impact it's ability to accomplish current and future assignments. The obvious example is a mobility problem which prevents a unit from getting to the required location. A more complex example is the degradation of a unit's warfare capability (e.g. the AAW capability of an aegis cruiser) may seriously degrade the overall capabilities of the battlegroup to which the unit is assigned. FRESH was partially successful at high-lighting situations which affected a future assignment of an individual unit.

| Event | Condition | Action |
|---|---|---|
| SORTS Received | If reported readiness does not meet requirements for current assignments. | a) Report deficiencies. |
| | If reported readiness does not meet requirements for future assignments. | a) Report deficiencies. |
| | If reported readiness degrades the aggregate requirements of a superior group. | a) Report deficiencies. |

**Figure 18 ECA SORT Report - Schedule Impacts**

Updates to Positional Databases

There are a variety of situations that could be detected by analysis of positional updates and movement reports submitted by units. A position report that shows a unit "out of position" to complete it's assigned

mission has already been discussed. Other possibilities include the possibility that the ship is standing into danger (e.g. deteriorating weather, a reported minefield in an area of increased tensions).

| Event | Condition | Action |
|---|---|---|
| Position Report Received | If reported position not consistent with current movement plan. | a) Report inconsistency.<br><br>b) Advise ship to update movement report. |

Figure 19 ECA Position Reported vs. Planned Movement

| Event | Condition | Action |
|---|---|---|
| Position Report or Movement Report Received | If reported position or planned movement indicates ship sailing into danger. | a) Report potential Hazard to operator.<br><br>b) Report potential hazard to ship.<br><br>c) Recommend new ship routing around hazard. |

Figure 20 ECA Position Reported vs. Potential Hazard

Updates to Schedule Databases

A change in schedule may result in insufficient units assigned to complete a scheduled event, too many units assigned to a scheduled event, the mismatch of unit capabilities with event requirements, or degrade the overall capability of a group of ships.

| Event | Condition | Action |
|---|---|---|
| Schedule Change Received | If schedule change results in failed or missed commitment. | a) Report failure to operator. |

Figure 21 ECA Schedule Change - Missed Commitment

UNCLASSIFIED
15

## Situational Planning

Situational Planni. g differs significantly from Situation Monitoring and Database Consistency/Integrity in applications of ADB concepts. The database would consist of operation and contingency plans which would be created and maintained with a variety of planning tools. Situational Planning events would not be defined as electronic updates to a database which could be evaluated automatically. An event which would affect Situational Planning would occur external to the plan database and be "defined as an event" for the database by an operator. Figure 1 portrays this.

The CASES planning tool is an excellent example of an application whose use can be triggered by external events. The external events can be either hypothetical (e.g., "What if North Korea invades South Korea") or actual (e.g., "Iraq has invaded Kuwait"). The definition of the external event can be used to search the existing plan library for a applicable plan.

A plan can be conceptualized as a collection of objects. Some objects are lower level plans which address specific parts of the plan, while others represent the resources which can be applied to the execution of the plan. An example of a plan as a collection of objects is shown if figure 22.
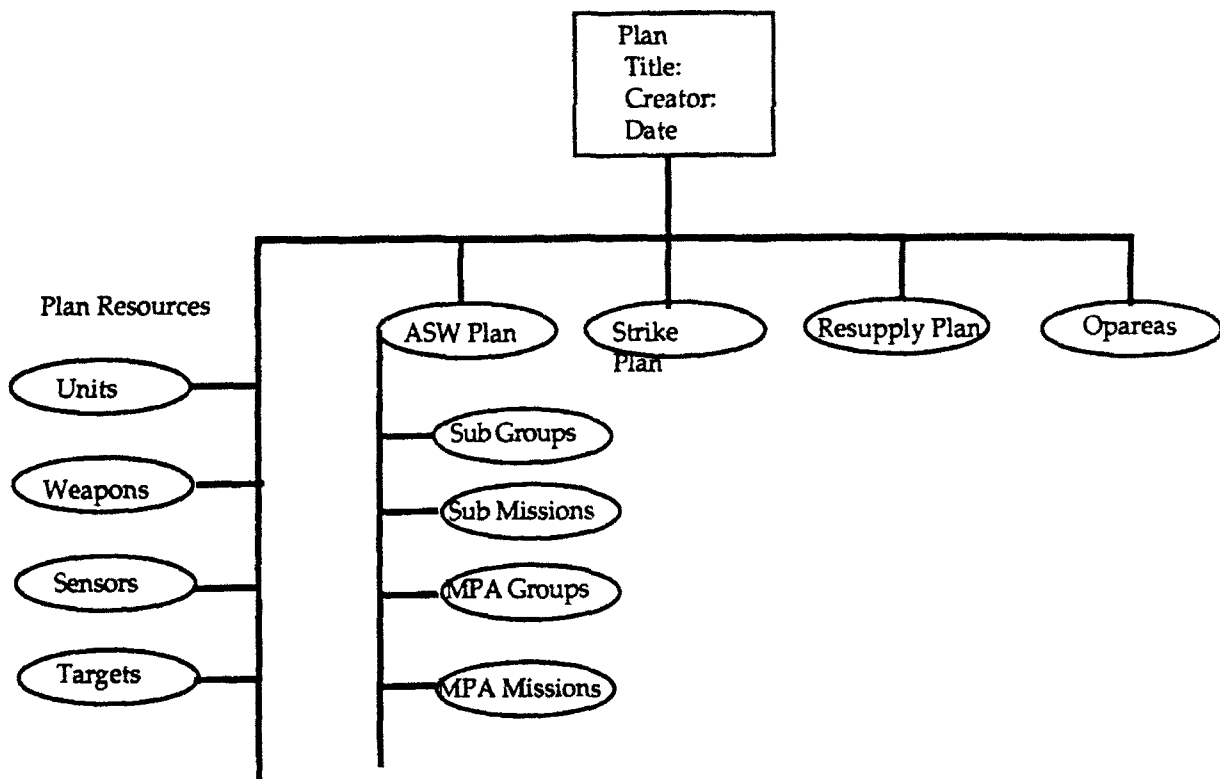
**Figure 22 A Plan as a collection of objects**

## Plan applicability

It is almost an axiom that you always have to fight the war you hadn't planned on. This varies from having a plan that is no longer applicable or only partially applicable to not having planned for a contingency at all. In the normal course of events a large number of plans are prepared and maintained which are never activated. With computer based planning tools these plans can be maintained in a library which can be searched automatically. Additionally, computer-based planning offers the planner the opportunity to examine and store multiple versions of the same scenario with different assumptions and courses of action (COAs).

Matching situation vs. plan assumptions

The structure of the electronic plan library needs to be conducive to evaluating the applicability of both the overall plan and of plan components and sub-components. In most cases no one plan will meet all the assumptions and requirements of a situation, however individual components may match the current situation very well. The ability to

build a new plan from the components of existing plans would speed up the planning process in a time critical environment. The first step would be to define the external event in terms of everything known or currently estimated including:

Political Alliances

Geography

Threat

    Types and numbers of weapons missiles, aircraft, ships, tanks

    qualitative assessment (e.g. High, Medium, Low) of warfare areas (e.g. MIW threat, AAW threat, ASW threat)

    Anticipated Actions/Response

Resources available to counter threat

    Types and numbers of weapons, missiles, aircraft, ships, tanks

TimeLine
*Expected Sequence*
*Expected Time*

Identify Courses of Action

Match warfare requirements to warfare components of existing plans

Exact definition of the situation in terms of assumptions is probably neither possible nor desired since it may constrain the search in a way which eliminates plans which do not match exactly but may accomplish desired objectives. This is particularly true in the category of available resources.

Appendix A includes the definition of a CASES plan and each of the components of a CASES plan. It is easy to see that plans can be extremely complex.

Plan feasibility

Updates to Order of Battle (OOB)

During peacetime operations, updates to OOBs take the form of intelligence reports which estimate current enemy capability and readiness reports which update the current capability of assigned units. During periods of hostility, unit damage reports and estimates of hostile losses would provide more time-critical impacts.

| Event | Condition | Action |
|-------|-----------|--------|
| Enemy OOB changes. | If change affects critical component and reflects an increase greater than 10%. | a) Report Changes. |

**Figure 23 ECA Order of Battle Changes**

Feasibility of existing operation and contingency plans.

Each operations plan or contingency plan is formulated with a base set of assumptions concerning the expected hostile OOB and the available resources which can be assigned to accomplish the goals of the plans. In most situations the feasibility of a plan is not dependent on a specific unit being available, however in some situations vital capabilities are only available in a limited number of units. The feasibility of a plan is certainly affected by significant changes in the expected hostile OOB or in available resources. Significance of a changes in OOB or available resources could be defined by a series of thresholds which when met might trigger different actions or reports.

## Air Traffic Control

Air Traffic Control (ATC) is a task performed by both the military and civilian agencies. ATC is an example of a highly dynamic database receiving hundreds of updates every minute. The application of ADB concepts to an ATC database is very attractive, however it is probably limited in the near future by stringent speed and accuracy requirements. Some possible applications of ADB concepts within the realm of ATC are listed below.

### Situation Monitoring

Situation Monitoring within the ATC environment might be divided into two categories, Safety of Flight and ATC System management. Safety of Flight monitoring combines all of the complexities of track correlation discussed above with the most stringent speed and accuracy and is probably not a good candidate for the application of ADB concepts in the near future. ATC System Management has much less stringent requirements of speed and is an excellent candidate for the application of ADB concepts. A few of the areas within ATC System Management are discussed below.

### Traffic Route Load

The ATC traffic route system is similar to the highway system, it has primary routes which are the most direct, cost effective route between two points and a variety of alternate routes which connect the same two destinations. The monitoring of traffic route loading allows the ATC to reroute traffic to avoid delays and congestion. There are a variety of inputs which need to be monitored to predict traffic loading including current position reports and flight plan filings.

| Event | Condition | Action |
|-------|-----------|--------|
| Flight Plan Filed | If proposed traffic exceeds traffic thresholds. | a) Report Threshold Violation. |

**Figure 24 ECA Flight Plan Filed - Exceeds Traffic Load Threshold**

## Airport Load

A primary contributor to changes in route loading is the loading of departure and destination airports. The constraints to airport capacity include the approach landing systems, number of runways, number of gates, the assignment of takeoff slots, and the overriding factor which affects all of the others, the current and future weather. Many of the other factors are affected by the surrounding ATC environment, e.g. the proximity of other airports, and the thresholds associated with each constraint may change as the result of conditions in the surrounding environment.

| Event | Condition | Action |
|---|---|---|
| Position Report received | If expected arrival time results in exceeding the airport capacity. | a) Report anticipated overload. |
| Gate Departure delay reported | If delay results in gate requirements exceeding gate capacity | a) Report anticipated overload. |

**Figure 25 ECA Airport Capacity**

## Navigational Aid Status

Navigational aids define the routes that make up the ATC system, each route segment is defined by two navigational aids placed to allow aircraft to always be in contact with at least one and preferably two radio beacons to ensure accurate navigation. The failure of a navigation aid can result in a section of the routing system being closed to air traffic which would require the re-routing of flights.

| Event | Condition | Action |
|---|---|---|
| Navigation Aid failure/unreliability reported | If flights currently enroute to navigation aid. | a) Identify flights.<br>b) Recommend re-routing. |

| | If current flights plans include navigation aid. | a) Identify flight plans.<br><br>b) Recommend re-routing. |
|---|---|---|

Figure 26 ECA Navigation Aid Failure

The installation of Global Positioning System (GPS) in civilian and military aircraft provides tremendous flexibility to the ATC routing system. The majority of aircraft will continue to rely on the network of navigational aids, however the ability to quickly identify those aircraft that can continue to navigate safely notwithstanding radio navigation aid failures could improve overall system safety. ATC controllers would be able to quickly differentiate and prioritize between flights that require immediate assistance and flights that could continue with onboard navigation.

| Event | Condition | Action |
|---|---|---|
| Navigation Aid failure/unreliability reported | If flights currently enroute to navigation aid. | a) Identify flights equipped with alternate navigation systems.<br><br>b) Identify flights not equipped with alternate navigation systems. |
| | If current flights plans include navigation aid. | a) Identify flight plans.<br><br>b) Identify flights equipped with alternate navigation systems.<br><br>c) Recommend re-routing of flights NOT equipped with alternate navigation systems. |

Figure 27 ECA Navigation Aid Failure - GPS Impact

## Flight Plan Feasibility

When a flight plan is filed it is based on the latest information available to the flight crew and proposes at departure time and expected arrival time at destination. Changing conditions at the departure airport, along the proposed route, and at the destination airport can invalidate the flight plan between the time it is filed and the actual take-off from the airport.

| Event | Condition | Action |
|---|---|---|
| Flight Plan filed with expected departure time.<br><br>At appropriate time intervals (e.g. 2 minutes) until takeoff report received. | If departure airport loading exceeds thresholds. | a) Identify flight plans affected.<br><br>b) Update expected departure times.<br><br>c) Recalculate expected arrival times. |
|  | If route loading or navigational aids fail. | a) Identify flight plans affected.<br><br>b) Recommend re-routing. |
|  | If destination airport loading exceeds thresholds. | a) Identify flight plans affected.<br><br>b) Determine valid arrival time.<br><br>c) Recalculate expected departure times. |

Figure 28 ECA Flight Plan Feasibility

## Situational Planning

Situational Planning in the ATC system could be used to develop a library of plans which would address major disruptions to the ATC system (e.g. closure of airports due to weather or accident). A library of plans could be created which would address potential major airport closures, the ADB system would then search the plan library looking for plans which would address the existing scenario. The search of the plan library could be

triggered by an airport closure message or by an externally defined event similar to that discussed in the section on Navy C2.

## Recommendations for a prototype implementation

### Navy Command and Control

DB Consistency/Integrity - Implementation of a prototype which addressed DB Consistency/Integrity would be constrained to an off-line demonstration addressing only a very small portion of the existing Navy C2 database. A successful prototype would almost certainly have no immediate impact on the future development of the OSS database due to potential problems with scaleability and performance.

Situation Monitoring - A successful prototype of situation monitoring has been demonstrated as part of the FRESH system. Some features of that successful prototype are already scheduled for implementation in the OSS database, although still as an external application.

Situational Planning - An ADB prototype addressing Navy C2 situational planning would have several advantages. The basic problem is relatively small scale compared to the OSS database and is not as constrained by real-time performance requirements. A prototype which addressed plan applicability and matched situation to plan assumptions could be implemented without impacting the performance of other systems. A successful prototype could have an immediate impact on situational planning capability currently being developed and installed at operational command centers.

### Air Traffic Control

Situation Monitoring - An ADB prototype addressing any of the possible applications to ATC databases would be constrained to an off-line, small scale application with little potential impact on the ATC system in the near or mid term. The current ATC database system is both very fragmented and highly constrained by real-time performance requirements. These two factors restrict ATC ADB prototypes to research for the foreseeable future.

Situational Planning - The development of a prototype to address ATC situational planning is precluded by the lack of any type of electronic situational planning database within the ATC system.

# Appendix A CASES Object Definitions

```
type CASES_Object = 520
    abbrev is cases
    subtype of Values
    annotation "defines cantypes common to all Cronus managers used by the Capabilities Assessment Expert
System (CASES)";

/****** General Enumeration Types ******/

cantype CASESASSUMPTIONGROUP
    representation is CasesAssumptionGroup:
    {StwAssumptionGroup = 1, AswAssumptionGroup = 2,
    AawAssumptionGroup = 3, NoAssumptionGroup = 4};

cantype CASESALLIANCE
    representation is CasesAlliance:
    {Friendly =1, Hostile = 2, Neutral =3, UnknownAlliance = 0};

cantype CASESCOLORCODE
    representation is CasesColorCode:
    {Blue = 1, Red = 2, Orange = 3, Green = 4, Yellow = 5,
    Cyan = 6, Brown = 7, White = 8, Black = 9, UnspecifiedColor = 0};

cantype CASESLANDBASETYPE
    representation is CasesLandBaseType:
    {Seaport = 1, Airfield = 2, SeaportOrAirfield = 3,
    UnknownLandBaseType = 0};

cantype CASESSEASON
    representation is CasesSeason:
    {Winter = 1, Spring = 2, Summer = 3,    Autumn= 4, UnknownSeason = 0};

cantype CASESSEASTATE
    representation is CasesSeaState:
    {Ss0 = 0, Ss1 = 1, Ss2    = 2, Ss3        = 3, Ss4 = 4,
    Ss5 = 5, Ss6 = 6, NoSs= 7};

cantype CASESWEATHER
    representation is CasesWeather:
    {Clear = 1, Overcast = 2, Rain = 3, FreezingRain = 4,
    Snow = 5, UnknownWeather = 0};

cantype CASESWINDSPEED
    representation is CasesWindSpeed:
    {Calm = 1, Freshening = 2, Squall = 3, Hurricane = 4,
    UnknownWindSpeed = 0};

cantype CASESOBJECTSTATUS
    representation is CasesObjectStatus:
    {Inactive =1, AtPort = 2, OnStation = 3, InTransit = 4,
    BetweenStates = 5, Completed = 6 , Dead = 7,
    UnknownObjectState = 0};
```

```
/****** Logistics Related Enumeration Types ******/

cantype CASESTRANSPORTTYPE
    representation is CasesTransportType:
    {TruckTransport = 1, RailTransport = 2, AirTransport = 3,
    SeaTransport = 4, OtherTransportType = 5, NoTransportType = 0};

cantype CASESSUPPLYCATEGORY
    representation is CasesSupplyCategory:
    {Wet = 1, Dry = 2, Ammo = 3, NoResupplyCategory = 0};

cantype CASESBACKGROUNDCONSUMPTIONTYPE
    representation is CasesBackgroundConsumptionType:
    {PerPersonPerDay      = 1, PerDay = 2, NoConsumption = 3,
    UnknownConsumption = 0};

cantype CASESRESUPPLYROLE
    representation is CasesResupplyRole:
    {Carrier = 1, Combtant = 2, SupplyShip = 3, Port = 4, NoResupplyrole = 0};

cantype CASESSUPPLYHANDLING
    representation is CasesSupplyHandling:
    {Crane = 1, SpecialCrane = 2, Pump = 4, SpecialPump = 8,
    Rack = 16, SpecialRack = 32, OtherSpecialHandling = 64, NoSpecialHandling = 0};

cantype CASESUNITOFMEASURE
    representation is CasesUnitOfMeasure:
    {Count = 1, Pounds = 2, Gallons = 3, Tons = 4, Feet = 5,
    SquareFeet = 6, CubicFeet = 7, NoMeasure = 0};

cantype CASESWEAPONTYPE
    representation is CasesWeaponType:
    {MpaTorpedo = 1, SubTorpedo = 2, Sm1Missile = 3, Sm2Missile = 4, Duck = 5,
    TlamC = 6, TlamD = 7, SpecialTlam = 8, AsuwMissile = 9,
    ArmMissile = 10, AirDecoy = 11, AsmMissile = 12, AamMissile = 13,
    SpecialAam = 14, SpecialSam = 15, SpecialAsm = 16,
    SpecialWeaponA = 17, SpecialWeaponB = 18, SpecialWeaponC = 19,
    OtherWeapon = 0};

/****** Force Activity Enumeration Types ******/

cantype CASESRAIDPROFILE
    representation is CasesRaidProfile:
    {SubSonic = 1, SuperSonic = 2, MixedRaidProfile = 3,
    OtherRaidProfile = 0};

cantype CASESAIRCRAFTROLE
    representation is CasesAircraftRole:
    {FighterEscort = 1, JammerEscort = 2, CarrierBasedAttack = 3,
    LandBasedAttack = 4, DecoyLauncher = 5, ArmLauncher = 6,
    AirborneTanker = 7, CombatAirPatrol = 8, DeckLaunchedInterceptor = 9,
    AawReserve = 10, StwReserve = 11, AsuwReserve = 12,
    AwacsRole = 13, MpaRole = 14, OtherAircraftRole = 0};
```

cantype CASESFORCEGROUPTYPE
    representation is CasesForceGroupType:
    {SubGroup = 1, MpaGroup = 2, StrikeGroup = 3,
    RaidGroup = 4, SagGroup = 5, OtherGroup = 6,
    UnknownForceGroupType = 0};

cantype CASESAIRCRAFTCATEGORY
    representation is CasesAircraftCategory:
    {A6 = 1, A7 = 2, F14 = 3, F15 = 4, F16 = 5, Fa18 = 6, Ea6b = 7,
    Kc10 = 8, Kc135 = 9, P3 = 10, S3 = 11, Lamps = 12,
    Awacs = 13, AirTransportCategory = 14,         Stealth = 15, OtherBomber = 16,
    OtherFighter = 17, OtherAircraftCategory = 0};

cantype CASESSHIPCATEGORY
    representation is CasesShipCategory:
    {submarine = 1, carrier = 2, SurfaceCombatant = 3, Resupply = 4,
    SurtassShip = 5, Tender = 6, PatrolCraft = 7, OtherShipCategory = 0};

cantype CASESAAWCAPABILITY
    representation is CasesAawCapability:
    {Sm1 = 1, Sm2 = 2, Aegis = 3, NoAawCapability = 0};

/****** ASW Related Enumeration Types ******/

cantype CASESSUBMARINEROLE
    representation is CasesSubmarineRole:
    {AreaPatrol = 1, BarrierPatrol = 2, GeneralPatrol = 3,
    SpecialPatrol = 4, OtherSubmarineRole = 0};

cantype CASESCUEINGSENSORTYPE
    representation is CasesCueingSensorType:
    {Sosus = 1, Surtass = 2, SpecialIuss = 3,
    LowFreqActive = 4, SpecialLfa = 5,
    OtherCueingSensor = 0};

cantype CASESSUBMARINEACTIVITY
    representation is CasesSubmarineActivity:
    {SubPatrol = 1, AreaSearch = 2, BarrierSearch = 3, SpaSearch = 4,
    SubTransit = 5, SubTrail = 6, SubLostTrail = 7, OtherSubmarineActivity = 0};

cantype CASESMPAACTIVITY
    representation is CasesMpaActivity:
    {InReserve = 1, Ingress = 2, MpaOnStation = 3, Egress = 4, Maintenance = 5,
    MpaTrail = 6, MpaLostTail = 7, OtherMpaActivity = 0};

cantype CASESSPATYPE
    representation is CasesSpaType:
    {BearingLine = 1, BearingBox = 2, Ellipse = 3, NoSpaType = 0};

cantype CASESSUBMISSIONTYPE
    representation is CasesSubMissionType:
    {AreaPatrolMission = 1, BarrierPatrolMission = 2, Transit = 3, ShipAttack = 4,

A-4

```
        NoSubMissionType = 0};

cantype CASESMPAMISSIONTYPE
    representation is CasesMpaMissionTpye:
    {MpaAreaSearch = 1, MpaBarrierSearch = 2, NoMpaMissionType = 0};

cantype CASESMINEMISSIONTYPE
    representation is CasesMineMissionType:
    {AswMineBarrier =1 , AswMineArea = 2,
    AsuwMineBarrier = 3, AsuwMineArea = 4,
    NoMineMissionType =0};

cantype CASESSUBMARINEBEHAVIOR
    representation is CasesSubmarineBehavior:
    {RandomWalk = 1, LadderWalk = 2, NoSubBehavior = 3,
    UnknownSubBehavior = 0};

/****** operation types for "edit-spec" ops *****/

cantype CASESRWTYPE
    representation is CasesRwType:
    {RwParameterSet = 1,
    RwResultSet = 2,
    RwResupplyItem = 3,
    RwResupplyFacility = 4,
    RwResupplyOperation = 5,
    RwResupplyDefs = 6,
    RwSourceLevelProfile = 7,
    RwSelfNoiseProfile = 8,
    RwPropLossCurve = 9,
    RwGeoDefaults = 10,
    RwTargetList = 11,
    RwTargetDeck = 12,
    RwTowedArray = 13,
    RwSonobuoy = 14,
    RwCueingSensor = 15,
    RwSensors = 16,
    RwTorpedo = 17,
    RwAirDelivered = 18,
    RwAawMissile = 19,
    RwAawDecoy = 20,
    RwSpecialWeapon = 21,
    RwWeapons = 22,
    RwMaritimePatrolClass = 23,
    RwAirCombatantClass = 24,
    RwSubsurfaceClass = 25,
    RwSurfaceCombatantClass = 26,
    RwAircraftCarrierClass = 27,
    RwResupplyShipClass = 28,
    RwShipClass = 29,
    RwAirClass = 30,
    RwClasses = 31,
    RwMpaUnit = 32,
```

```
RwCombatantAirUnit = 33,
RwAirUnit = 34,
RwSubmarine = 35,
RwSurfaceCombatant = 36,
RwAircraftCarrier = 37,
RwResupplyShip = 38,
RwShip = 39,
RwUnits = 40,
RwAswArea = 41,
RwAswBarrier = 42,
RwAswTransit = 43,
RwStrikeOparea = 44,
RwAirRaidOparea = 45,
RwSagOparea = 46,
RwResupplyOparea = 47,
RwBomberWave = 48,
RwPort = 49,
RwOpareas = 50,
RwSubMission = 51,
RwMpaMission = 52,
RwMpaExclusionZone = 53,
RwMineMission = 54,
RwSubGroup = 55,
RwMpaGroup = 56,
RwMineGroup = 57,
RwSubResults = 58,
RwMpaResults = 59,
RwMineResults = 60,
RwAswPlan = 61,
RwStwMission = 62,
RwAirRaidMission = 63,
RwSagMission = 64,
RwCarrierGroup = 65,
RwStwSupportGroup = 66,
RwAirRaidGroup = 67,
RwSagGroup = 68,
RwStwResults = 69,
RwStwPlan = 70,
RwResupplyMission = 71,
RwResupplyGroup = 72,
RwResupplyUnitResult = 73,
RwResupplyPlan = 74,
RwPlan = 75,
RwSubAttackMission = 76,
UndefinedRwType = 0};
```

/****** Basic Object Cantypes ******/

```
cantype CASESSECURITYLABEL
    representation is CronusCasesSecurityLabel:
    record
    Level: ASC;
    CompartmentsAndCaveats: array of ASC;
```

```
        Comments: ASC;
        end CASESSECURITYLABEL;

cantype CASESLOCATION
        representation is CronusCasesLocation:
        record
        DegLat:             F32                    annotation "Latitude in decimal degrees - south
                                                   negative";
        DegLon:             F32                    annotation "Longitude in decimal degrees - west
                                                   negative";
        end CasesLocation;


cantype CASESSPACELOCATION
        representation is CronusCasesSpaceLocation:
        record
        MapCoordinates:     CASESLOCATION;
        Altitude:           F32;
        end CasesSpaceLocation;


cantype CASESITEMQUANTITY
        representation is CronusCasesItemQuantity:
        record
        ItemName:           ASC                    annotation "A string indicating the type or name of
                                                   the item";
        Quantity:           F32                    annotation "The quantity or value of the item";
        end CasesItemQuantity;


cantype CASESITEMTABLE
        representation is CronusCasesItemTable:
        record
        Name:               ASC                    annotation "A string identifier for the table";
        Items:              array of CASESITEMQUANTITY   annotation "A list of items and thier
                                                   quantities";
        end CasesItemTable;

/* a near-term implementation, soon to be replaced by "values" mechanism */

cantype CASESPARAMETER
        representation is CronusCasesParameter:
        record
        Name:               ASC;
        RowLabels:          array of ASC;
        ColLabels:          array of ASC;
        StringValues:       array of ASC;
        NumericValues:      array of F32;
        end CasesParameter;


cantype CASESPARAMETERSET
        representation is CronusCasesParameterSet:
        record
        Name:               ASC;
        GroupName:          ASC;
        Creator:            ASC;
```

```
    ValuesFlag:          EBOOL                              annotation "Toggles between using Values or
                                                            Parameters";
    ValuesData:          array of EUID;
    Parameters:          array of CASESPARAMETER;
    end CasesParameterSet;


cantype CASESRESULT
    representation is CronusCasesResult:
    record
    Name:                ASC;
    RowLabels:           array of ASC;
    ColLabels:           array of ASC;
    NumericValues:       array of F32;
    end CasesResult;


cantype CASESRESULTSET
    representation is CronusCasesResultSet:
    record
    Name:                ASC;
    ValuesFlag:          EBOOL                              annotation "Toggles between using Values or
                                                            Outcomes";
    ValuesData:          array of EUID;
    Outcomes:            array of CASESRESULT;
    end CasesResultSet;



/****** Basic Resupply Cantypes ******/

cantype CASESRESUPPLYITEM
    representation is CronusCasesResupplyItem:
    record
    Name:                ASC                                annotation "A string identifier for this supply item";
    WetDryAmmo:          CASESSUPPLYCATEGORY                annotation "Indicates supply category as Wet, Dry
                                                            or Ammo";
    Consumption:         CASESBACKGROUNDCONSUMPTIONTYPE     annotation "The type of
                                                            background consumption calculation";
    Requirements:        array of CASESSUPPLYHANDLING       annotation "Indicates handling requirements
                                                            for this supply item";
    LoadPriority:        S32I                               annotation "Lower values indicate this item is
                                                            loaded before others";
    Measure:             CASESUNITOFMEASURE                 annotation "The unit of measure to be used for this
                                                            supply item";
    UnitWeight:          F32                                annotation "Weight of an individual item, in tons, if
                                                            appropriate";
    end CasesResupplyItem;


cantype CASESRESUPPLYFACILITY
    representation is CronusCasesResupplyFacility:
    record
    Name:                ASC                                annotation "The name of this type of facility";
    Capabilities:        array of CASESSUPPLYHANDLING       annotation "Indicates supply handling
                                                            capabilities of this type of facility";
```

```
WetPerDay:          array of F32          annotation "Tons of wet supplies movable per day
                                          as a function of sea-state":

DryPerDay:          array of F32          annotation "Tons of dry supplies movable per day
                                          as a function of sea-state";

AmmoPerDay:         array of F32          annotation "Tons of ammo supplies movable per
                                          day as a function of sea-state";

    end CasesResupplyFacility;


cantype CASESRESUPPLYOPERATION
    representation is CronusCasesResupplyOperation:
    record
    Name:               ASC;
    Facilities:         array of ASC      annotation "Names of facility types available for
                                          this operation";

    StartupTime:        F32               annotation "Typical start-up time for this operation,
                                          in days";

    CompletionTime:     F32               annotation "Typical completion time for this
                                          operation, in days";

    end CasesResupplyOperation;


cantype CASESRESUPPLYDEFS
    representation is CronusCasesResupplyDefs:
    record
    Items:              array of CASESRESUPPLYITEM;
    Facilities:         array of CASESRESUPPLYFACILITY;
    Operations:         array of CASESRESUPPLYOPERATION;
    end CasesResupplyDefs;


cantype CASESRESUPPLYSETTING
    representation is CronusCasesResupplySetting:
    record
    ItemName:           ASC               annotation "A name of a resupply item used by a
                                          unit";

    OnHand:             F32               annotation "How many or how much of the item the
                                          unit has on hand";

    StockageObjective:  F32               annotation "Inventory level for item when unit is
                                          considered full";

    BasicLoad:          F32               annotation "Inventory level unit must maintain for
                                          its own use (not give away)";

    ReorderLevel:       F32               annotation "Inventory level at which unit will
                                          requisition more of the item";\

    MaxCarry:           F32               annotation "Max amount of item unit can carry if
                                          that was all it was carrying";

    Consumption:        F32               annotation "Amount of item unit consumes per-day
                                          or per-person-per-day";

    SpecialData:        array of F32      annotation "A place to record special data
                                          associated with this supply setting";

    end CasesResupplySetting;


/* Status of resupply characteristics can change as a function of damage, time at sea, etc */


cantype CASESSUPPLYSTATUS
    representation is CronusCasesSupplyStatus:
```

```
record
FromTime:          F32                    annotation "Start of simulated time interval for this
                                          status, or zero for initial";
ToTime:            F32                    annotation "End of simulated time interval for this
                                          status, or zero for initial";
Settings:          array of CASESRESUPPLYSETTING  annotation "Status of each resupply item at
                                          this time interval";
SpecialInfo:       array of F32           annotation "Place to record information particular to
                                          a given unit, etc";
end CasesSupplyStatus;


/****** Basic ASW Cantypes ******/

cantype CASESSOURCELEVEL
     representation is CronusCasesSourceLevel:
     record
     LowerSpeed:        F32               annotation "The lower speed for this source level
                                          profile";
     UpperSpeed:        F32               annotation "The upper speed for this source level
                                          profile";
     Frequency:         F32               annotation "The frequency for this source level
                                          profile";
     SourceLevel:       S32I              annotation "The source level value in decibels";
     end CasesSourceLevel;

cantype CASESSOURCELEVELPROFILE
     representation is CronusCasesSourceLevelProfile:
     record
     Name:              ASC;
     Profile:           array of CASESSOURCELEVEL;
     end CasesSourceLevelProfile;

cantype CASESSELFNOISE
     representation is CronusCasesSelfNoise:
     record
     LowerSpeed:        F32               annotation "The lower speed value for this self
                                          noise profile";
     UpperSpeed:        F32               annotation "The upper speed value for this slef
                                          noise profile";
     SelfNoise:         S32I              annotation "The self noise value in decibels";
     end CasesSelfNoise;

cantype CASESSELFNOISEPROFILE
     representation is CronusCasesSelfNoiseProfile:
     record
     Name:              ASC;
     Profile:           array of CASESSELFNOISE;
     end CasesSelfNoiseProfile;

cantype CASESPROPLOSSCURVE
     representation is CronusCasesPropLossCurve:
     record
```

```
    FirstRange:          F32                  annotation "The range (in nm) of the first prop-loss
                                              value";
    RangeSampling:       F32                  annotation "The sampling interval (in nm) of each
                                              prop-loss value";
    PropLossValues:      array of F32         annotation "The prop-loss in dB at each sample
                                              interval";
    end CasesPropLossCurve;


cantype CASESGEOCELL
    representation is CronusCasesGeoCell:
    record
    Location:            CASESLOCATION        annotation "The actual geographic location of this
                                              cell";
    MappedFlag:          EBOOL                annotation "Indicates if this cell is mapped to a
                                              different cell";
    MappedLoc:           CASESLOCATION        annotation "The location of the cell that this cell is
                                              mapped to, if any";
    AmbNoiseFlag:        EBOOL                annotation "If true, then ambient noise value
                                              overrides that in database";
    AmbNoiseVal:         F32;
    PropLossFlag:        EBOOL                annotation "If true, then prop-loss curve overrides
                                              that in database";
    PropLossCurve:       CASESPROPLOSSCURVE;
    end CasesGeoCell;


cantype CASESGEODEFAULTS
    representation is CronusCasesGeoDefaults:
    record
    Creator:             ASC;
    Title:               ASC;
    Comment:             array of ASC;
    Cells:               array of CASESGEOCELL;
    end CasesGeoDefaults;


/****** Target Related Cantypes ******/


cantype CASESSORTIETYPE
    representation is CronusCasesSortieType:
    record
    AircraftType:        ASC                  annotation "Type of aircraft flown for this sortie";
    WeaponType:          ASC                  annotation "Type of weapon carried for this sortie";
    WeaponCount:         S32I                 annotation "Number of weapons carried for this
                                              sortie";
    end CasesSortieType;


cantype CASESAIMPOINT
    representation is CronusCasesAimpoint:
    record
    Name:                ASC                  annotation "A string identifier for this aimpoint";
    SspdValues:          array of F32         annotation "An SSPD value for each sortie type";
    end CasesAimpoint;


cantype CASESTARGET
```

representation is CronusCasesTarget:
record

| | | |
|---|---|---|
| Name: | ASC | annotation "A string identifier for this target, not necessarily unique"; |
| Specializer: | ASC | annotation "Cat-Code relation to Cases target class hierarchy"; |
| Location: | CASESLOCATION | annotation "Location of this target"; |
| CountryCode: | ASC | annotation "Country code to which this target belongs"; |
| CatCode: | ASC | annotation "DOD Category code describing this target"; |
| BeNumber: | ASC | annotation "Unique DOD identifier for this target"; |
| DatabaseId: | S32I | annotation "Unique database identifier for this target"; |
| DatabaseList: | ASC | annotation "Database list to which this target belongs, if any"; |
| Radius: | F32 | annotation "Effective radius of target, if applicable"; |
| Requirements: | array of ASC | annotation "Things required for this target to operate, if any"; |
| Dependencies: | array of ASC | annotation "Things this target provides that are required by other targets"; |
| Functional: | EBOOL | annotation "Flag indicating if target is considered functional"; |
| DestructionTime: | F32 | annotation "If non-functional, time when target was destroyed"; |
| Aimpoints: | array of CASESAIMPOINT | annotation "Individual aimpoints that comprise this target"; |
| RelatedPorts: | array of ASC | annotation "Names of ports affected by the state of this target"; |

end CasesTarget;

cantype CASESTARGETLIST
representation is CronusCasesTargetList:
record

| | | |
|---|---|---|
| Name: | ASC | annotation "A string identifier for this target list"; |
| Alliance: | CASESALLIANCE | annotation "Identifies targets as belonging to friendly, enemy or neutral forces"; |
| SortieTypes: | array of CASESSORTIETYPE | annotation "An ordered list of sortie type definitions"; |
| BeNumbers: | array of ASC | annotation "A list of targets, by Be-Number"; |
| TargetCenter: | CASESLOCATION | annotation "A place to record a representative location"; |

end CasesTargetList;

cantype CASESTARGETDECK
representation is CronusCasesTargetDeck:
record

| | | |
|---|---|---|
| Name: | ASC | annotation "A string identifier for this target deck"; |
| Alliance: | CASESALLIANCE | annotation "Identifies targets as belonging to friendly, enemy or neutral forces"; |

```
SortieTypes:          array of CASESSORTIETYPE  annotation "Sortie definitions for aimpoint
                                                weaponeering data";
Targets:              array of CASESTARGET      annotation "Target objects that comprise this target
                                                deck";
TargetCenter:         CASESLOCATION             annotation "A place to record a representative
                                                location";
end CasesTargetDeck;
```

/***** Sensor Cantypes *****/

```
cantype CASESGENERICSENSORINFO
    representation is CronusCasesGenericSensorInfo:
    record
    Class:              ASC         annotation "Class name from the equipment
                                    hierarchy, or a notional class name";
    InheritsFrom:       ASC         annotation "Class this sensor is based on, if this is a
                                    notional sensor";
    Type:               ASC         annotation "Type node from the equipment
                                    hierarchy";
    Category:           ASC         annotation "An even less-specific node from the
                                    equipment hierarchy";
    SupplyCategory:     ASC         annotation "The name of a supply category for this
                                    sensor";
    DirectivityIndex:   S32I        annotation "The directivity index characteristic of
                                    this sensor";
    RecDifferential:    S32I        annotation "The recognition differential
                                    characteristic of this sensor";
    end CasesGenericSensorInfo;

cantype CASESTOWEDARRAY
    representation is CronusCasesTowedArray:
    record
    GenericInfo:        CASESGENERICSENSORINFO    annotation "Generic info for a towed array
                                                  sensor";
    SelfNoiseProfiles:  array of CASESSELFNOISEPROFILE  annotation "The self-noise generated by this
                                                  array at various speeds";
    end CasesTowedArray;

cantype CASESSONOBUOY
    representation is CronusCasesSonobuoy:
    record
    GenericInfo:        CASESGENERICSENSORINFO    annotation "Generic info for an expendable
                                                  sensor";
    ExpendedSearch:     S32I        annotation "The quantity of sensors expended per
                                    search pattern";
    ExpendedLoc:        S32I        annotation "The quantity of sensors expended per
                                    localization effort";
    ExpendedHourly:     S32I        annotation "The qunatity of sensors expended per
                                    hour while trailing target";
    end CasesSonobuoy;

cantype CASESCUEINGSENSOR
    representation is CronusCasesCueingSensor:
```

A-13

```
record
Name:              ASC;
Type:              CASESCUEINGSENSORTYPE              annotation "The (enumeration) type of this
                                                      sensor";
Identifier:        S32I                               annotation "A unique (numerical) identifier for this
                                                      sensor";
Location:          CASESLOCATION                      annotation "The sensor location throughout the
                                                      simulation";
Orientation:       S32I                               annotation "The sensor compass heading throughout
                                                      the simulation";
ActiveParams:      array of F32                       annotation "Special parameters for active cueing
                                                      elements";

end CasesCueingSensor;


cantype CASESSENSORS
      representation is CronusCasesSensors:
      record
      TowedArrays:       array of CASESTOWEDARRAY;
      Sonobuoys:         array of CASESSONOBUOY;
      Cueing:            array of CASESCUEINGSENSOR;
      end CasesSensors;


/****** Weapon Cantypes ******/


cantype CASESGENERICWEAPONINFO
      representation is CronusCasesGenericWeaponInfo:
      record
      Class:             ASC                          annotation "Class name from the equipment
                                                      hierarchy, or a notional class name";
      InheritsFrom:      ASC                          annotation "Class this sensor is based on, if this is a
                                                      notional sensor";
      Type:              ASC                          annotation "Type node from the equipment
                                                      hierarchy";
      Category:          ASC                          annotation "An even less-specific node from the
                                                      equipment hierarchy";
      CasesType:         CASESWEAPONTYPE              annotation "The generic Cases enumeration type";
      SupplyCategory:    ASC                          annotation "The name of a supply category for this
                                                      weapon";
      Expended:          S32I                         annotation "The quantity of weapons expended per
                                                      engagement";
      Pk:                F32                          annotation "The default Pk associated with this
                                                      weapon per engagement";
      Range:             F32                          annotation "The range of this weapon (in nm)";
      Speed:             F32                          annotation "The speed of this weapon (in fps)";
      Weight:            F32                          annotation "The weight of a single weapon (in
                                                      tons)";

end CasesGenericWeaponInfo;

cantype CASESTORPEDO
      representation is CronusCasesTorpedo:
      record
      GenericInfo:       CASESGENERICWEAPONINFO       annotation "Generic info for a torpedo";
```

```
    AirDeliverable:       EBOOL                                annotation "Indicates if this type of torpedo can be
                                                               carried by MPA units";
    SubDeliverable:       EBOOL                                annotation "Indicates if this type of torpedo can be
                                                               carried by subsurface units";
    AtSeaTransferable:    EBOOL                                annotation "Indicates if this type of torpedo can be
                                                               resupplied at sea";
    end CasesTorpedo;

cantype CASESAIRDELIVERED
    representation is CronusCasesAirDelivered:
    record
    GenericInfo:          CASESGENERICWEAPONINFO     annotation "Generic info for an air-delivered
                                                               weapon";
    AssocWeapon:          ASC                                  annotation "Name of associated weapon (kits), if
                                                               any";
    JettisonFlag:         EBOOL                                annotation "Indicates if weapons are jettisoned
                                                               when aircraft in difficulty";
    BomberPk:             F32                                  annotation "Pk for weapon against bombers";
    FighterPk:            F32                                  annotation "Pk for weapon against fighters";
    SamPk:                F32                                  annotation "Pk for weapon against Sams";
    end CasesAirDelivered;

cantype CASESAAWMISSILE
    representation is CronusCasesAawMissile:
    record
    GenericInfo:          CASESGENERICWEAPONINFO     annotation "Generic info for a SM1 or
                                                 SM2";
    end CasesAawMissile;

cantype CASESAAWDECOY
    representation is CronusCasesAawDecoy:
    record
    GenericInfo:          CASESGENERICWEAPONINFO     annotation "Generic info for a duck";
    end CasesAawDecoy;

cantype CASESSPECIALWEAPON
    representation is CronusCasesSpecialWeapon:
    record
    GenericInfo:          CASESGENERICWEAPONINFO     annotation "Generic info for a special
                                                 weapon of some sort";
    end CasesSpecialWeapon;

cantype CASESWEAPONS
    representation is CronusCasesWeapons:
    record
    Torpedos:             array of CASESTORPEDO;
    AirDelivered:         array of CASESAIRDELIVERED;
    AawMissiles:          array of CASESAAWMISSILE;
    AawDecoys:            array of CASESAAWDECOY;
    SpecialWeapons:       array of CASESSPECIALWEAPON;
    end CasesWeapons;

/****** Class Cantypes ******/
```

A-15

```
cantype CASESGENERICCLASSINFO
      representation is CronusCasesGenericClassInfo:
      record
      Class:                ASC            annotation "Class name from the unit hierarchy, or a
                                           notional class name";
      InheritsFrom:         ASC            annotation "Class this class is based on, if this is a
                                           notional class";
      Type:                 ASC            annotation "Type node from the unit hierarchy";
      Category:             ASC            annotation "An even less-specific node from the
                                           unit hierarchy";
      DefaultSpeed:         F32            annotation "Usual transit or patrol speed (in kts)";
      MaxSpeed:             F32            annotation "Maximum burst speed (in kts)";
      MaxSustSpeed:         F32            annotation "Maximum sustainable speed (in kts)";
      FuelCap:              F32            annotation "Fuel capacity (in lbs) of POL type fuel,
                                           if appropriate";
      FuelConsumption:      F32            annotation "Reasonable fuel consumption: lbs/day
                                           for ships, lbs/nm for aircraft";
      Range:                F32            annotation "Range without refueling (twice the
                                           unrefueled radius for aircraft";
      MaintenanceTime:      F32            annotation "Average time required for maintenance
                                           between missions";
      Crew:                 S32I           annotation "Typical number of crew members (ship
                                           or squadron compliment)";
      ReorderInterval:      F32            annotation "Days betwen requisitions for
                                           background-consumed supplies";
      Supplies:             array of       CASESRESUPPLYSETTING        annotation
                                           "Loadouts & resupply data on weapons, sensors,
                                           etc";
      SpecialData:          array of F32   annotation "A place for special characteristics and
                                           other types of data";
      SpecialTables:        array of CASESITEMTABLE   annotation "A place for special data to be stored in
                                           table format";
      end CasesGenericClassInfo;


cantype CASESMARITIMEPATROLCLASS
      representation is CronusCasesMaritimePatrolClass:
      record
      GenericInfo:          CASESGENERICCLASSINFO annotation "Generic info for a Maritime Patrol
                                           Aircraft class";
      CasesType:            CASESAIRCRAFTCATEGORY;
      MinStationTime:       F32            annotation "Minimum time that aircraft will be
                                           scheduled to be on station";
      MaxStationTime:       F32            annotation "Maximum time aircraft can remain on
                                           station";
      MaxTransitTime:       F32            annotation "Maximum enroute transit time";
      end CasesMaritimePatrolClass;


cantype CASESAIRCOMBATANTCLASS
      representation is CronusCasesAirCombatantClass:
      record
      GenericInfo:          CASESGENERICCLASSINFO annotation "Generic info for an Air Combatant
                                           class";
```

```
CasesType:              CASESAIRCRAFTCATEGORY;
TakeoffAbort:           F32                          annotation "Probability this class aborts takeoff
                                                     (from deck or runway)";
AirAbort:               F32                          annotation "Probability this class aborts mission in
                                                     air";
DownSquawk:             F32                          annotation "Probability this class aborts due to
                                                     downsquawk";
RepairParams:           array of F32                 annotation "Obscure parameters describing repair
                                                     characteristics of this class";
Jettison:               F32                          annotation "Probability weapons will be jettisoned
                                                     during SAM avoidance";
WildFire:               F32                          annotation "Probability pilot will fire wild during a
                                                     dogfight";
AamExpend:              F32                          annotation "Average AAMs expended per
                                                     engagement";
HomeBaseType:           CASESLANDBASETYPE            annotation "Denotes aircraft carrier, airfield or
                                                     tanker base";
StealthFactor:          F32                          annotation "A multiplier for stealth-related
                                                     characteristics";
PossibleRoles:          array of CASESAIRCRAFTROLE   annotation "A list of roles this particular
                                                     class can have";
end CasesAirCombatantClass;


cantype CASESSUBSURFACECLASS
     representation is CronusCasesSubsurfaceClass:
     record
     GenericInfo:            CASESGENERICCLASSINFO   annotation "Generic info for a Submarine class";
     CasesType:              CASESSHIPCATEGORY;
     SourceLevels:          array of ASC            annotation "Names of source level profiles for this
                                                     class";
     end CasesSubsurfaceClass;


cantype CASESSURFACECOMBATANTCLASS
     representation is CronusCasesSurfaceCombatantClass:
     record
     GenericInfo:            CASESGENERICCLASSINFO   annotation "Generic info for a Surface Combatant
                                                     class";
     CasesType:              CASESSHIPCATEGORY;
     Embarkable:             EBOOL                   annotation "Indicates whether helicoptors can be
                                                     embarked on this class";
     AawCapability:          CASESAAWCAPABILITY      annotation "Indicates highest AAW capability for
                                                     this class";
     end CasesSurfaceCombatantClass;


cantype CASESAIRCRAFTCARRIERCLASS
     representation is CronusCasesAircraftCarrierClass:
     record
     GenericInfo:            CASESGENERICCLASSINFO   annotation "Generic info for an Aircraft Carrier
                                                     class";
     CasesType:              CASESSHIPCATEGORY;
     CanCarry:               array of CASESAIRCRAFTCATEGORY   annotation "The general types of
                                                     aircraft that can be carried";
     end CasesAircraftCarrierClass;
```

```
cantype CASESRESUPPLYSHIPCLASS
    representation is CronusCasesResupplyShipClass:
    record
    GenericInfo:        CASESGENERICCLASSINFO  annotation "Generic info for a Supply Ship class";
    CasesType:          CASESSHIPCATEGORY;
    Capabilities:       array of CASESSUPPLYHANDLINGannotation "The types of resupply
                                                   capabilities this class has";
    Operations:         array of ASC             annotation "The types of resupply operations this
                                                   class can perform";
    end CasesResupplyShipClass;

cantype CASESSHIPCLASS
    representation is CronusCasesShipClass:
    record
    GenericInfo:        CASESGENERICCLASSINFO  annotation "Generic info for a Ship Super-Category
                                                   class ";
    CasesType:          CASESSHIPCATEGORY;
    end CasesShipClass;

cantype CASESAIRCLASS
    representation is CronusCasesAirClass:
    record
    GenericInfo:        CASESGENERICCLASSINFO  annotation "Generic info for an Air Super-Category
                                                   class";
    CasesType:          CASESAIRCRAFTCATEGORY;
    end CasesAirClass;

cantype CASESCLASSES
    representation is CronusCasesClasses:
    record
    Mpa:                array of CASESMARITIMEPATROLCLASS;
    AirCombatant:       array of CASESAIRCOMBATANTCLASS;
    Subsurface:         array of CASESSUBSURFACECLASS;
    SurfaceCombatant:   array of CASESSURFACECOMBATANTCLASS;
    AircraftCarrier:    array of CASESAIRCRAFTCARRIERCLASS;
    ResupplyShip:       array of CASESRESUPPLYSHIPCLASS;
    OtherShips:         array of CASESSHIPCLASS;
    OtherAir:           array of CASESAIRCLASS;
    end CasesClasses;


/****** Unit Cantypes ******/

cantype CASESUNITSTATE
    representation is CronusCasesUnitState:
    record
    FromDay:            F32                      annotation "Simulation start time of snapshot, in
                                                   days";
    UntilDay:           F32                      annotation "Simulation end time of snapshot, in
                                                   days";
    Location:           CASESLOCATION            annotation "Actual (or inferred) unit location";
```

| | | |
|---|---|---|
| Status: | CASESOBJECTSTATUS | annotation "General state of object as enumerated type"; |
| DamageLevel: | F32 | annotation "An indication of damage sustained so far"; |
| MissionProfile: | F32 | annotation "Number of days out of port, sorties flown, etc"; |
| SupplyLevels: | CASESITEMTABLE | annotation "Number of weapons and other supplies aboard"; |
| Comment: | array of ASC | annotation "A place to record significant comments, etc"; |
| SpecificData: | array of F32 | annotation " A place to record specific state information as desired"; |

end CasesUnitState;

cantype CASESGENERICUNITINFO
    representation is CronusCasesGenericUnitInfo:
    record

| | | |
|---|---|---|
| Class: | ASC | annotation "A class name from the unit hierarchy"; |
| Type: | ASC | annotation "A type name from the unit hierarchy"; |
| Category: | ASC | annotation "A node name from the unit hierarchy, higher than type"; |
| Flag: | ASC | annotation "Code of country to which unit belongs"; |
| Mfg: | ASC | annotation "Code of country that manufactures unit, based on class"; |
| Alliance: | CASESALLIANCE | annotation "Specifies unit as friendly, hostile or neutral"; |
| Hull: | ASC | annotation "A unique hull number for this unit"; |
| Name: | ASC | annotation "A unique name for this unit"; |
| ReorderInterval: | F32 | annotation "May either override class-level attribute, or be set to zero"; |
| SupplyStatus: | array of CASESSUPPLYSTATUS | annotation "May either override class-level attribute, or be set to nil"; |
| HomePort: | ASC | annotation "A name of a port or airfield"; |
| ForceGroup: | ASC | annotation "Name of force group his unit belongs to"; |
| SimulatedStates: | array of CASESUNITSTATE | annotation "A place to record state transitions during a single simulation"; |

end CasesGenericUnitInfo;

cantype CASESMPAUNIT
    representation is CronusCasesMpaUnit:
    record

| | | |
|---|---|---|
| GenericInfo: | CASESGENERICUNITINFO | annotation "Describes the generic unit characteristics"; |
| Platforms: | CASESITEMTABLE | annotation "Classes and counts of actual aircraft in the squadron"; |
| Embarkation: | ASC | annotation "Name of unit this unit is embarked on (overrides allocations)"; |

end CasesMpaUnit;

cantype CASESCOMBATANTAIRUNIT
    representation is CronusCasesCombatantAirUnit:
    record

```
        GenericInfo:         CASESGENERICUNITINFO    annotation  "Describes   the   generic   unit
                                                     characteristics";
        Platforms:           CASESITEMTABLE          annotation "Classes and counts of actual aircraft in
                                                     the squadron or regiment";
        Embarkation:         ASC                     annotation "Name of unit this unit is embarked on
                                                     (overrides allocations)";

        end CronusCasesCombatantairUnit;

cantype CASESAIRUNIT
        representation is CronusCasesAirUnit:
        record
        GenericInfo:         CASESGENERICUNITINFO    annotation  "Describes   the   generic   unit
                                                     characteristics";
        Platforms:           CASESITEMTABLE          annotation "The classes and counts of aircraft in the
                                                     squadron or regiment ";
        Embarkation:         ASC                     annotation "Name of unit this unit is embarked on
                                                     (overrides allocations)";

        end CasesAirUnit;

cantype CASESSUBMARINE
        representation is CronusCasesSubmarine:
        record
        GenericInfo:         CASESGENERICUNITINFO    annotation  "Describes   the   generic   unit
                                                     characteristics";

        end CasesSubmarine;

cantype CASESSURFACECOMBATANT
        representation is CronusCasesSurfaceCombatant:
        record
        GenericInfo:         CASESGENERICUNITINFO    annotation  "Describes   the   generic   unit
                                                     characteristics";
        EmbarkedUnits:       array of ASC            annotation "The list of embarked air units by name,
                                                     if any";

        end CasesSurfaceCombatant;

cantype CASESAIRCRAFTCARRIER
        representation is CronusCasesAircraftCarrier:
        record
        GenericInfo:         CASESGENERICUNITINFO    annotation  "Describes   the   generic   unit
                                                     characteristics";
        EmbarkedUnits:       array of ASC            annotation "The list of embarked air units by name,
                                                     if any";

        end CasesAircraftCarrier;

cantype CASESRESUPPLYSHIP
        representation is CronusCasesResupplyShip:
        record
        GenericInfo:         CASESGENERICUNITINFO    annotation  "Describes   the   generic   unit
                                                     characteristics";

        end CasesResupplyShip;

cantype CASESSHIP
        representation is CronusCasesShip:
```

```
                record
                GenericInfo:           CASESGENERICUNITINFO;
                end CasesShip;

        cantype CASESUNITS
                representation is CronusCasesUnits:
                record
                Mpa:                   array of CASESMPAUNIT;
                CombatantAir:          array of CASESCOMBATANTAIRUNIT;
                OtherAir:              array of CASESAIRUNIT;
                Submarines:            array of CASESSUBMARINE;
                SurfaceCombatants:     array of CASESSURFACECOMBATANT;
                Carriers:              array of CASESAIRCRAFTCARRIER;
                SupplyShips:           array of CASESRESUPPLYSHIP;
                OtherShips:            array of CASESSHIP;
                end CasesUnits;


        /****** Oparea Cantypes ******/

        cantype CASESASWAREA
                representation is CronusCasesAswArea:
                record
                Name:          ASC                      annotation "A string identifier for this area, unique
                                                        to plan";

                Flags:         array of ASC             annotation "Country codes of those who patrol this
                                                        Area";

                Alliance:      CASESALLIANCE            annotation "Identifies patrollers as friendly, enemy
                                                        or neutral";

                Vertices:      array of CASESLOCATION   annotation "Geographical coordinates of the Area
                                                        boundaries";

                end CasesAswArea;


        cantype CASESASWBARRIER
                representation is CronusCasesAswBarrier:
                record
                Name:          ASC                      annotation "A string identifier for this barrier,
                                                        unique to plan";

                Flags:         array of ASC             annotation "Country codes of those who know
                                                        about this Barrier";

                Alliance:      CASESALLIANCE            annotation "Identifies barrier as friendly or enemy";
                Endpoints:     array of CASESLOCATION   annotation "Geographical coordinates of the Barrier
                                                        endpoints";

                end CasesAswBarrier;

        cantype CASESASWTRANSIT
                representation is CronusCasesAswTransit:
                record
                Name:          ASC                      annotation "A string identifier for this transit track,
                                                        unique to plan";

                Flags:         array of ASC             annotation "Country codes of those who transit
                                                        along this track";

                Alliance:      CASESALLIANCE            annotation "Identifies transitors as friendly, enemy
                                                        or neutral";
```

| Waypoints: | array of CASESLOCATION | annotation "Geographical coordinates of track waypoints"; |
| Radius: | F32 | annotation "Radius within which transitors must pass through waypoints"; |

end CasesAswTransit;

cantype CASESSPA
    representation is CronusCasesSpa:
    record

| Name: | ASC; |
| Type: | CASESSPATYPE; |
| SpaTime: | F32; |
| Target: | ASC; |
| Searcher: | ASC; |
| CenterPoint: | CASESLOCATION; |
| MajorAxis: | F32; |
| MinorAxis: | F32; |
| Orientation: | F32; |
| SpaPosition: | CASESLOCATION; |
| SpaLength: | F32; |
| HalfWidth: | F32; |
| EndPoint: | CASESLOCATION; |
| Bearing: | F32; |
| SpaWidth: | F32; |

    end CasesSpa;

cantype CASESSTRIKEOPAREA
    representation is CronusCasesStrikeOparea:
    record

| Name: | ASC | annotation "A string identifier for this configuration, unique to plan"; |
| Flags: | array of ASC | annotation "Country codes of those participating in strike"; |
| Alliance: | CASESALLIANCE | annotation "Identifies striking forces as friendly or enemy"; |
| Location: | CASESLOCATION | annotation "Represents geographic center of the battle force"; |
| CarrierRadius: | F32 | annotation "Represents dispersion of carriers, or zero if not more than one CV"; |
| CapRadius: | F32 | annotation "Represents maximum extent of combined combat air patrol"; |
| AswRadius: | F32 | annotation "Represents extent of ASW area to be patrolled"; |
| AsuwRadius: | F32 | annotation "Represents extent of ASUW area to be patrolled"; |
| LbtBases: | array of ASC | annotation "Names of airfields from which tanker aircraft may fly"; |
| LbaBases: | array of ASC | annotation "Names of airfields from which attack aircraft may fly"; |
| AawBases: | array of ASC | annotation "Names of airfields from which AAW support aircraft may fly"; |
| TargetList: | ASC | annotation "Name of target list that identifies the targets to be struck"; |

A-22

```
TargetCenter:        CASESLOCATION              annotation "Location of the computed center-of-
                                                mass of the targets";

end CasesStrikeOparea;

cantype CASESAIRRAIDOPAREA
     representation is CronusCasesAirRaidOparea:
     record
     Name:            ASC                       annotation "A string identifier, unique to plan";
     Flags:           array of ASC              annotation "Country codes of those participating in
                                                raids";
     Alliance:        CASESALLIANCE             annotation 'Identifies raiding forces as friendly or
                                                enemy";
     Airfields:       array of ASC              annotation "Names of the Airfields from which
                                                raiding aircraft will fly";
     Waypoints:       array of CASESLOCATION    annotation "A set of waypoints that the aircraft must
                                                pass througn";
     AirfieldCenter:  CASESLOCATION             annotation "Location of center of mass of the
                                                airfields";
     Profile:         CASESRAIDPROFILE          annotation "The attack profile the raiders will use";
     end CasesAirRaidOparea;

cantype CASESSAGOPAREA
     representation is CronusCasesSagOparea:
     record
     Name:            ASC                       annotation "A string identifier, unique to plan";
     Flags:           array of ASC              annotation "Country codes of those participating in
                                                SAG";
     Alliance:        CASESALLIANCE             annotation "Identifies SAG as friendly or enemy";
     Location:        CASESLOCATION             annotation "Represents geographic center of the
                                                SAG";
     SensorRadius:    F32                       annotation "Represents range at which battle group
                                                could be detected";
     ThreatRadius:    F32                       annotation "Represents range within which battle
                                                group sould be attacked";
     SpecialTables:   array of CASESITEMTABLE   annotation "A place for special asuw parameters";
     end CasesSagOparea;

cantype CASESRESUPPLYOPAREA
     representation is CronusCasesResupplyOparea:
     record
     Name:            ASC                       annotation "A string identifier for this resupply
                                                SLOC, unique to plan";
     Flags:           array of ASC              annotation "Country codes of those participating in
                                                resupply";
     Alliance:        CASESALLIANCE             annotation "Identifies participants as friendly or
                                                enemy";
     FromPorts:       array of ASC              annotation "Names of Ports from which supplies are
                                                retrieved";
     ToPorts:         array of ASC              annotation "Names of Ports to which supplies are
                                                delivered, if any";
     ToStwOpareas:    array of ASC              annotation "Names Strike Opareas to which
                                                supplies are delivered, if any";
```

```
SpecialTables:        array of CASESITEMTABLE   annotation "A place for special logistics resupply
                                                parameters";
end CasesResupplyOparea;

cantype CASESPORT
    representation is CronusCasesPort:
    record
    Name:             ASC;
    Flags:            array of ASC;
    Alliance:         CASESALLIANCE;
    Type:             CASESLANDBASETYPE;
    Location:         CASESLOCATION;
    Personnel:        S32I                       annotation "Personnel used in background
                                                consumption calculations";
    ReorderInterval:  F32                        annotation "Days between re-ordering background
                                                consumption items";
    Operations:       array of ASC               annotation "Types of simultaneous resupply
                                                operations this port can perform";
    SupplyStatus:     array of CASESSUPPLYSTATUS  annotation "Status and characteristics of
                                                each supply item handled by this port";
    InitialUnits:     array of ASC               annotation "Names of units initially located at this
                                                port";
    SupplyRecords:    array of CASESITEMTABLE    annotation "Records of supply levels during
                                                simulations";
    Active:           EBOOL                      annotation "Flag indicating if port is still active
                                                during a simulation";
    DayLost:          F32                        annotation "Day this port became inactive during
                                                simulation";
    end CasesPort;

cantype CASESBOMBERWAVE
    representation is CronusCasesBomberWave:
    record
    Name:             ASC;
    Regiments:        array of ASC               annotation "List of regiments scheduled to attack in
                                                this wave";
    Delay:            F32                        annotation "Number of minutes delay from pervious
                                                wave";
    NumAxes:          S32I                       annotation "Number of threat axes the aircraft will
                                                distribute themselves over";
    Interceptors:     array of ASC               annotation "Names of airfields from which
                                                interceptors are to fly, if any";
    ReturnTo:         ASC                        annotation "Name of airfield regiments are to return
                                                to after mission is complete";
    end CasesBomberWave;

cantype CASESOPAREAS
    representation is CronusCasesOpareas:
    record
    AswArea:          array of CASESASWAREA;
    AswBarrier:       array of CASESASWBARRIER;
    AswTransit:       array of CASESASWTRANSIT;
    Strike:           array of CASESSTRIKEOPAREA;
```

```
AirRaid:              array of CASESAIRRAIDOPAREA;
BomberWave:           array of CASESBOMBERWAVE;
Sag:                  array of CASESSAGOPAREA;
Resupply:             array of CASESRESUPPLYOPAREA;
Port:                 array of CASESPORT;
end CasesOpareas;
```

/****** ASW Missions; the Plan and its Results ******/

```
cantype CASESSUBMISSION
      representation is CronusCasesSubMission:
      record
      Name:           ASC:
      Comment:        array of ASC;
      GroupName:      ASC              annotation "Name of the submarine force group
                                       performing this mission":
      OpareaName:     ASC              annotation "Name of an area, barrier, transit or
                                       strike oparea";
      CasesType:      CASESSUBMISSIONTYPE   annotation "Indetifies mission as area, barrier,
                                       transit, or ship attack";
      PriorMission:   ASC              annotation "Name of mission just prior to this
                                       mission, if any";
      NextMission:    ASC              annotation "Name of mission right after this
                                       mission, if any";
      StartDay:       F32              annotation "The day the mission is scheduled to
                                       begin";
      EndDay:         F32              annotation "The day the mission is scheduled to
                                       end";
      Duration:       F32              annotation "The duration of the mission in days":
      PatrolSpeed:    F32              annotation "The speed of the searching
                                       submarines";
      PatrolBehavior: CASESSUBMARINEBEHAVIOR   annotation "Indicates type of search pattern
                                       as random or ladder-walk";
      SourceLevels:   array of ASC     annotation "Names of expected source-level profiles
                                       to search for";
      TargetSpeed:    F32              annotation "The expected speed of the target
                                       submarines";
      end CasesSubMission;

cantype CASESSUBATTACKMISSION
      representation is CronusCasesSubAttackMission:
      record
      Name:           ASC;
      Comment:        array of ASC;
      GroupName:      ASC              annotation "Name of the submarine force group
                                       performing this mission";
      OpareaName:     ASC              annotation "Name of an area oparea where subs
                                       patrol while waiting to attack";
      CvMissionName:  ASC              annotation "Name of the carrier mission
                                       representing the attack";
      CasesType:      CASESSUBMISSIONTYPE   annotation "Indetifies mission as area, barrier,
                                       transit, or ship attack";
```

```
    PriorMission:        ASC                        annotation "Name of mission just prior to this
                                                    mission, if any";
    NextMission:         ASC                        annotation "Name of mission right after this
                                                    mission, if any";
    StartDay:            F32                        annotation "The day the mission is scheduled to
                                                    begin";
    EndDay:              F32                        annotation "The day the mission is scheduled to
                                                    end";
    Duration:            F32                        annotation "The duration of the mission in days";
    PatrolSpeed:         F32                        annotation "The speed of the searching
                                                    submarines";

    end CasesSubAttackMission;

cantype CASESMPAMISSION
    representation is CronusCasesMpaMission:
    record
    Name:                ASC;
    Comment:             array of ASC;
    GroupName:           ASC                        annotati  "Name of the force group perfcming the
                                                    mission";
    OpareaName:          ASC                        annotation "Name of an area or barrier oparea";
    BaseName:            ASC                        annotation "Name of the land base from which the
                                                    Mpa units will fly";
    CasesType:           CASESMPAMISSIONTYPE        annotation "Indetifies mission as area or barrier
                                                    search";
    StartDay:            F32                        annotation "The day the mission is scheduled to
                                                    begin";
    EndDay:              F32                        annotation "The day the mission is scheduled to
                                                    end";
    Duration:            F32                        annotation "The duration of the mission in days";
    SourceLevels:        array of ASC               annotation "Names of expected source-level profiles
                                                    to search for";
    TargetSpeed:         F32                        annotation "The expected speed of the target
                                                    submarines";

    end CasesMpaMission;

cantype CASESMPAEXCLUSIONZONE
    representation is CronusCasesMpaExclusionZone:
    record
    Name:                ASC                        annotation "Name of this exclusion zone, unique to
                                                    plan";
    Comment:             array of ASC;
    AreaName:            ASC                        annotation "Name of the ASW Area describing
                                                    alliance & geometry";
    StartDay:            F32                        annotation "Day at which Mpa are to be excluded
                                                    from this zone";
    EndDay:              F32                        annotation "Day at which exclusion no longer
                                                    applies";
    Duration:            F32                        annotation "Total duration of exclusion status";
    end CasesMpaExclusionZone;

/* We have left room for Mines to be maintained by minelayers in the future...*/
```

```
cantype CASESMINEMISSION
      representation is CronusCasesMineMission:
      record
      Name:               ASC;
      Comment:            array of ASC;
      GroupName:          ASC                        annotation "Name of the force group doing the
                                                     mining";
      OpareaName:         ASC                        annotation "Name of a barrier oparea, perhaps areas
                                                     in the future as well";
      BaseName:           ASC                        annotation "Name of the land base from which
                                                     airborne minelayers will fly?";
      CasesType:          CASESMINEMISSIONTYPE       annotation "Indetifies type as ASW or ASUW area
                                                     or barrier";
      StartDay:           F32                        annotation "The day the mines are schduled to be in
                                                     place";
      EndDay:             F32                        annotation "The day the mines are scheduled to be
                                                     removed";
      Duration:           F32                        annotation "The number of days the mines are
                                                     scheduled to be in place";
      Mines:              CASESITEMTABLE             annotation "The types and quantities of the mines to
                                                     be maintained";

      end CasesMineMission;

cantype CASESSUBGROUP
      representation is CronusCasesSubGroup:
      record
      Name:               ASC                        annotation "Name of this group, unique to plan";
      Alliance:           CASESALLIANCE;
      Comment:            array of ASC;
      Units:              array of ASC               annotation "Names of submarine units allocated to
                                                     this group";
      Missions:           array of ASC               annotation "An ordered list of missions this group
                                                     will conduct";

      end CasesSubGroup;

cantype CASESMPAGROUP
      representation is CronusCasesMpaGroup:
      record
      Name:               ASC                        annotation "Name of this group, unique to plan";
      Alliance:           CASESALLIANCE;
      Comment:            array of ASC;
      Units:              array of ASC               annotation "Names of MPA squadrons allocated to
                                                     this group";
      Missions:           array of ASC               annotation "an ordered list of MPA missions this
                                                     group will conduct";

      end CasesMpaGroup;

cantype CASESMINEGROUP
      renresentation is CronusCasesMineGroup:
      record
      Name:               ASC                        annotation "Name of this group, unique to plan, -
                                                     not used yet";

      Alliance:           CASESALLIANCE;
```

```
Comment:            array of ASC;
Units:              array of ASC          annotation "Names of minelayer units allocated to
                                          this group - not used yet";
Missions:           array of ASC          annotation "Just a list of all the mine missions in the
                                          plan, for now";

    end CasesMineGroup;

cantype CASESSUBRESULTS
    representation is CronusCasesSubResults:
    record
    Presence:           CASESITEMTABLE;
    OnStation:          CASESITEMTABLE;
    TorpedosUsed:       CASESITEMTABLE;
    UnitsLost:          CASESITEMTABLE;
    Detections:         CASESITEMTABLE;
    CuedDetections:     CASESITEMTABLE;
    SubsUnderTrail:     CASESITEMTABLE;
    MeanTrailTime:      CASESITEMTABLE;
    Kills:              CASESITEMTABLE;
    end CasesSubResults;

cantype CASESMPARESULTS
    representation is CronusCasesMpaResults:
    record
    Availability:       CASESITEMTABLE;
    Sorties:            CASESITEMTABLE;
    OnStationDays:      CASESITEMTABLE;
    InFlightDays:       CASESITEMTABLE;
    MaintenanceDays:    CASESITEMTABLE;
    TorpedosUsed:       CASESITEMTABLE;
    AircraftLost:       CASESITEMTABLE;
    Detections:         CASESITEMTABLE;
    CuedDetections:     CASESITEMTABLE;
    SubsUnderTrail:     CASESITEMTABLE;
    MeanTrailTime:      CASESITEMTABLE;
    Kills:              CASESITEMTABLE;
    end CasesMpaResults;

cantype CASESMINERESULTS
    representation is CronusCasesMineResults:
    record
    Presence:           CASESITEMTABLE;
    MinesLost:          CASESITEMTABLE;
    Kills:              CASESITEMTABLE;
    end CasesMineResults;

cantype CASESASWPLAN
    representation is CronusCasesAswPlan:
    record
    SubGroups:          array of CASESSUBGROUP;
    SubMissions:        array of CASESSUBMISSION;
    AttackMissions:     array of CASESSUBATTACKMISSION;
    MpaGroups:          array of CASESMPAGROUP;
```

```
MpaMissions:        array of CASESMPAMISSION;
MineGroups:         array of CASESMINEGROUP;
MineMissions:       array of CASESMINEMISSION;
ExclusionZones:     array of CASESMPAEXCLUSIONZONE;
SubResults:         array of CASESSUBRESULTS;
MpaResults:         array of CASESMPARESULTS;
MineResults:        array of CASESMINERESULTS;
end CasesAswPlan;
```

/****** Strike Missions; the Plan & its Results          ******/

```
cantype CASESSTWMISSION
    representation is CronusCasesStwMission:
    record
    Name:               ASC;
    Comment:            array of ASC;
    GroupName:          ASC          annotation "Name of the battle force performing this
                                     mission";
    LbaGroupName:       ASC          annotation "Name of the Land-Based attack group
                                     supporting this mission";
    LbsGroupName:       ASC          annotation "Name of the Land-Based AAW support
                                     group for this mission";
    LbtGroupName:       ASC          annotation "Name of the Land-Based Tanker group
                                     supporting this mission";
    OpareaName:         ASC          annotation "Name of a strike oparea";
    PriorMission:       ASC          annotation "Name of mission just prior to this
                                     mission, if any";
    NextMission:        ASC          annotation "Name of mission right after this
                                     mission, if any";
    StartDay:           F32          annotation "The day the mission is scheduled to
                                     begin";
    EndDay:             F32          annotation "The day the mission is scheduled to
                                     end";
    MaxDuration:        F32          annotation "The maximum allowed duration of the
                                     mission, in days";
    MaxStrikes:         S32I         annotation "The maximum number of strikes
                                     allowed";
    NumTlam:            S32I         annotation "Number of TLAM allocated for use in
                                     these strikes";
    StwAssumptions:     array of ASC annotation "Names of parameter sets";
    AsuwAssumptions:    array of ASC annotation "Names of parameter sets";
    AswAssumptions:     array of ASC annotation "Names of parameter sets";
    AawAssumptions:     array of ASC annotation "Names of parameter sets";
    end CasesStwMission;

cantype CASESAIRRAIDMISSION
    representation is CronusCasesAirRaidMission:
    record
    Name:               ASC;
    Comment:            array of ASC;
    GroupName:          ASC          annotation "Name of the air group performing this
                                     mission";
```

A-29

```
RaidOpareaName:      ASC                 annotation "Name of an air raid oparea";
StwOpareaName:       ASC                 annotation "Name of a strike oparea";
PriorMission:        ASC                 annotation "Name of mission just prior to this
                                         mission, if any";
NextMission:         ASC                 annotation "Name of mission right after this
                                         mission, if any";
StartDay:            F32                 annotation "The day the mission is scheduled to
                                         begin";
EndDay:              F32                 annotation "The day the mission is scheduled to
                                         end";
MinUnits:            S32I                annotation "The minimum number of units required
                                         to perform this mission";
BomberWaves:         array of ASC        annotation    "Names of wave objects for this
                                         raid";
Assumptions:         array of ASC        annotation "Names of parameter sets";
end CasesAirRaidMission;

cantype CASESSAGMISSION
     representation is CronusCasesSagMission:
     record
     Name:               ASC;
     Comment:            array of ASC;
     GroupName:          ASC                 annotation "Name of the air group performing this
                                             mission";
     SagOpareaName:      ASC                 annotation "Name of a sag oparea";
     StwOpareaName:      ASC                 annotation "Name of a strike oparea";
     PriorMission:       ASC                 annotation "Name of mission just prior to this
                                             mission, if any";
     NextMission:        ASC                 annotation "Name of mission right after this
                                             mission, if any";
     StartDay:           F32                 annotation "The day the mission is scheduled to
                                             begin";
     EndDay:             F32                 annotation "The day the mission is scheduled to
                                             end";
     Assumptions:        array of ASC        annotation "Names of parameter sets";
     end CasesSagMission;

cantype CASESCARRIERGROUP
     representation is CronusCasesCarrierGroup:
     record
     Name:               ASC                 annotation "Name of this battle group, unique to
                                             plan";
     Alliance:           CASESALLIANCE;
     Comment:            array of ASC;
     Units:              array of ASC        annotation "Names of units allocated to this group";
     Missions:           array of ASC        annotation "An ordered list of strike missions this
                                             group will conduct";

     end CasesCarrierGroup;

cantype CASESSTWSUPPORTGROUP
     representation is CronusCasesStwSupportGroup:
     record
     Name:               ASC;
```

```
    Alliance:              CASESALLIANCE;
    Comment:               array of ASC;
    Units:                 array of ASC;
    Missions:              array of ASC          annotation "A list of Strike missions this group of
                                                 units will support";

    end CasesStwSupportGroup;

cantype CASESAIRRAIDGROUP
    representation is CronusCasesAirRaidGroup:
    record
    Name:                  ASC;
    Alliance:              CASESALLIANCE;
    Comment:               array of ASC;
    Units:                 array of ASC;
    Missions:              array of ASC;
    end CasesAirRaidGroup;

cantype CASESSAGGROUP
    representation is CronusCasesSagGroup:
    record
    Name:                  ASC;
    Alliance:              CASESALLIANCE;
    Comment:               array of ASC;
    Units:                 array of ASC;
    Missions:              array of ASC;
    end CasesSagGroup;

cantype CASESSTWRESULTS
    representation is CronusCasesStwResults:
    record
    Duration:              array of F32;
    Strikes:               array of F32;
    AircraftSorties:       array of CASESITEMTABLE;
    StwWpnsUsed:           array of CASESITEMTABLE;
    AawWpnsUsed:           array of CASESITEMTABLE;
    AswWpnsUsed:           array of CASESITEMTABLE;
    AsuwWpnsUsed:          array of CASESITEMTABLE;
    TargetsKilled:         array of CASESITEMTABLE;
    AimpointsKilled:       array of CASESITEMTABLE;
    SubsKilled:            array of CASESITEMTABLE;
    RaidersKilled:         array of CASESITEMTABLE;
    SagUnitsKilled:        array of CASESITEMTABLE;
    SupplyStatus:          array of CASESSUPPLYSTATUS;
    AcStwLosses:           array of CASESITEMTABLE;
    AcAawLosses:           array of CASESITEMTABLE;
    AcAswLosses:           array of CASESITEMTABLE;
    AcPercentages:         array of CASESITEMTABLE;
    ShipAawLosses:         array of CASESITEMTABLE;
    ShipAswLosses:         array of CASESITEMTABLE;
    ShipAsuwLosses:        array of CASESITEMTABLE;
    ShipPercentages:       array of CASESITEMTABLE;
    end CasesStwResults;
```

```
cantype CASESSTWPLAN
      representation is CronusCasesStwPlan:
      record
      CarrierGroups:        array of CASESCARRIERGROUP;
      CarrierMissions:      array of CASESSTWMISSION;
      SupportGroups:        array of CASESSTWSUPPORTGROUP;
      AirRaidGroups:        array of CASESAIRRAIDGROUP;
      AirRaidMissions:      array of CASESAIRRAIDMISSION;
      SagGroups:            array of CASESSAGGROUP;
      SagMissions:          array of CASESSAGMISSION;
      Results:              array of CASESSTWRESULTS;
      end CasesStwPlan;


/****** Resupply Missions; Plan and its Results        ******/


cantype CASESRESUPPLYMISSION
      representation is CronusCasesResupplyMission:
      record
      Name:             ASC;
      Comment:          array of ASC;
      GroupName:        ASC               annotation "Name of the units performing this
                                          mission";
      OpareaName:       ASC               annotation "Name of a resupply oparea";
      PriorMission:     ASC               annotation "Name of mission just prior to this
                                          mission, if any";
      NextMission:      ASC               annotation "Name of mission right after this
                                          mission, if any";
      StartDay:         F32               annotation "The day the mission is scheduled to
                                          begin";
      EndDay:           F32               annotation "The day the mission is scheduled to
                                          end";
      Assumptions:      array of ASC      annotation "Names of parameter sets";
      end CasesResupplyMission;


cantype CASESRESUPPLYGROUP
      representation is CronusCasesResupplyGroup:
      record
      Name:             ASC;
      Alliance:         CASESALLIANCE;
      Comment:          array of ASC;
      Units:            array of ASC;
      Missions:         array of ASC;
      end CasesResupplyGroup;


cantype CASESRESUPPLYITEMRESULT
      representation is CronusCasesResupplyItemResult:
      record
      ItemName:         ASC               annotation "Name of resupply item";
      Received:         array of F32      annotation "Average, 90th, 10th percentile of
                                          number of this item received";
      Used:             array of F32      annotation "Average, 90th, 10th percentile of
                                          number of this item used";
```

```
Transferred:          array of F32              annotation "Average, 90th, 10th percentile of
                                                number of this item transferred";
FinalOnHand:          array of F32              annotation "Average, 90th, 10th percentile of final
                                                items on hand";
     end CasesResupplyItemResult;

cantype CASESRESUPPLYUNITRESULT
     representation is CronusCasesResupplyUnitResult:
     record
     UnitName:             ASC;
     ItemResults:          array of CASESRESUPPLYITEMRESULT;
     end CasesResupplyUnitResult;

cantype CASESRESUPPLYPLAN
     representation is CronusCasesResupplyPlan:
     record
     Groups:               array of CASESRESUPPLYGROUP;
     Missions:             array of CASESRESUPPLYMISSION;
     SpecialTables:        array of CASESITEMTABLE annotation "A place to record special logistics
                                                information";
     Results:              array of CASESRESUPPLYUNITRESULT;
     end CasesResupplyPlan;

/****** The Plan Object ******/

cantype CASESPLAN
     representation is CronusCasesPlan:
     record
     Creator:              ASC;
     Title:                ASC;
     SecurityLabel:        CASESSECURITYLABEL;
     Comment:              array of ASC;
     GeographicField:      array of ASC              annotation "A list of countries invloved in the plan";
     ExerciseName:         ASC                       annotation "Name of the Fleet Exercise this plan
                                                represents";
     Oplan:                ASC                       annotation "Name of the Operation Plan this plan
                                                represents";
     OtherAttributes:      array of ASC;
     PriorPlans:           array of EUID;
     StartTime:            EDATE;
     LastEvalTime:         EDATE;
     LastEditTime:         EDATE;
     GeoLoc:               CASESGEODEFAULTS;
     SourceLevelProfiles:  array of CASESSOURCELEVELPROFILE;
     ParameterSets:        array of CASESPARAMETERSET;
     ResultSets:           array of CASESRESULTSET;
     ResupplyDefs:         CASESRESUPPLYDEFS;
     Sensors:              CASESSENSORS;
     Weapons:              CASESWEAPONS;
     Classes:              CASESCLASSES;
     Units:                CASESUNITS;
     TargetDecks:          array of CASESTARGETDECK;
     TargetLists:          array of CASESTARGETLIST;
```

```
Opareas:            CASESOPAREAS;
AswPlan:            CASESASWPLAN;
StrikePlan:         CASESSTWPLAN;
ResupplyPlan:       CASESRESUPPLYPLAN;
EvalOptions:        array of CASESPARAMETERSET;
SimulationMode:     array of CASESPARAMETERSET;
SpecialParams:      array of CASESPARAMETERSET;
SpecialTables:      array of CASESITEMTABLE;
UnitSummary:        array of CASESUNITS;
end CasesPlan;
```

/****** ERRORS AND WARNINGS ******/

error OPEN_PLAN_LIMIT_EXCEEDED
    message "Cannot open plan object - Limit on number of Open plans exceeded"
    returns(S32I);

error PLAN_OBJECT_NOT_FOUND
    message "Plan object cannot be found in local library"
    returns(EUID);

error PLAN_OBJECT_IN_USE
    message "Cannot open plan object - Locked by another user"
    returns(EUID);

error OVERLAYS_NOT_CHANGED
    message "Some of the selected objects are overlays and will not be changed"
    returns(ASC);

error POSSIBLE_CONFLICT
    message "Requested operation may result in undesired side-effects"
    returns(ASC);

error CANNOT_PERFORM_OPERATION
    message "Requested operation cannot be performed"
    returns(ASC);

end type CASES_Object;

# REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | May 1993 | Final |

**4. TITLE AND SUBTITLE**

DATA AND KNOWLEDGE ENGINEERING (DKE)
Case Study Report

**6. AUTHOR(S)**

BBN Systems and Technologies

**5. FUNDING NUMBERS**

C: N66001-90-D-0048
PE: 0602232N
WN: DN306242

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

BBN Systems and Technologies
4015 Hancock Street
San Diego, CA 92110

**8. PERFORMING ORGANIZATION REPORT NUMBER**

TR 2515

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152-5001

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT (Maximum 200 words)**

The purpose of this study is to identify potential for the application of Active Database (ADB) concepts to Navy Command and Control ($C^2$).

**14. SUBJECT TERMS**

| | |
|---|---|
| active/reactive database | situation monitoring |
| probability theory | situation planning |
| decision-making databases | air traffic control |

**15. NUMBER OF PAGES**

70

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | SAME AS REPORT |

| 21a. NAME OF RESPONSIBLE INDIVIDUAL | 21b. TELEPHONE (include Area Code) | 21c. OFFICE SYMBOL |
|---|---|---|
| Leah Y. Wong, COTR | (619) 553–4127 | Code 441 |